# PATENTABLE SUBJECT MATTER AFTER *ALICE*—DISTINGUISHING NARROW SOFTWARE PATENTS FROM OVERLY BROAD BUSINESS METHOD PATENTS

*Ognjen Zivojnovic*[†]

For the last forty years, software has perplexed patent law. On the one hand, all software consists of algorithms—a series of logical or arithmetic steps performed on loaded and stored numbers[1]—which courts have long considered patent-ineligible abstract ideas.[2] On the other hand, there is no dispute that the application of an abstract idea may be patent-eligible,[3] and physical machines programmed with software offer tangible improvements to our daily lives.[4] In fact, software is responsible for many of the most valuable advances of the twentieth and early twenty-first centuries.[5] Courts have repeatedly confronted the question of where, on a spectrum between a patent-ineligible abstract idea and a patent-eligible application thereof, a claim directed to a piece of software implemented by a computer falls, but their results have been inconsistent and confusing.[6] Thus, the central

[†] J.D., 2015, University of California, Berkeley, School of Law.

[1] *See* DAVID A. PATTERSON & JOHN L. HENNESSY, COMPUTER ORGANIZATION AND DESIGN: THE HARDWARE/SOFTWARE INTERFACE 76–78 (4th ed. 2009) (listing all instructions in the MIPS assembly language, i.e., all possible steps that an exemplary processor in a computer can execute); Timothy B. Lee, *Software Is Just Math. Really.*, FORBES (Aug. 11, 2011, 2:29 PM), http://www.forbes.com/sites/timothylee/2011/08/11/software-is-just-math-really/ (arguing that all software ever written boils down to a mathematical algorithm).

[2] *See* Gottschalk v. Benson, 93 U.S. 63, 71–73 (1972).

[3] Diamond v. Diehr, 450 U.S. 175, 187–88 (1981).

[4] *See* CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1305 (Fed. Cir. 2013) (en banc) (Rader, C.J., concurring in part and dissenting in part), *aff'd*, 134 S. Ct. 2347 (2014).

[5] *See id.*; PATTERSON, *supra* note 1, at 3–4.

[6] *Compare* CLS Bank Int'l v. Alice Corp. Pty. Ltd., 685 F.3d 1341, 1356 (Fed. Cir. 2012) (arguing that claims directed to software are only invalid if "it [is] manifestly evident that the claims are patent-ineligible under § 101"), *vacated*, 717 F.3d 1269 (Fed. Cir. 2013) (en banc), *aff'd*, 134 S. Ct. 2347 (2014) *with* Bancorp Servs. L.L.C. v. Sun Life Ins. Co. of Can. (U.S.), 687 F.3d 1266, 1278 (Fed. Cir. 2012) (arguing that "a computer must be integral to the claimed invention, facilitating the process in a way that

question plaguing software patents—what additional limitations are "enough"[7] to transform a patent-ineligible abstract idea into a patent-eligible application—remains unanswered.

In its most recent decision addressing the patent eligibility of software, *Alice Corp. Pty. Ltd. v. CLS Bank International*,[8] the Supreme Court held that adding the words "apply it with a computer" to a patent-ineligible abstract idea is not "enough" to confer patent eligibility.[9] This holding can be interpreted narrowly, affecting only business method patents—i.e., software patents that amount to little more than a fundamental economic practice (i.e., a "business method") applied "with a computer." However, *Alice* also endorsed a theory that the exceptions to patent eligibility (including the abstract idea exception) are substantive limitations meant to protect against overly board patents.[10] In the lower court en banc decision,[11] four Federal Circuit judges argued that adopting this substantive limitation theory would be the death knell for all software patents.[12] While the Supreme Court left ample room for interpretation between these two extremes, whether due to *Alice* or other factors, lower courts have invalidated the majority of software patents challenged under § 101 since the *Alice* decision.[13]

Few dispute that many of the recently invalidated software patents, especially business method patents, are more likely to burden than promote innovation.[14] In fact, commentators have argued that business

a person making calculations or computations could not" before it is enough to confer patent eligibility onto software).

[7] Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1297 (2012) ("To put the matter more precisely, do the patent claims add *enough* to their statement of [patent-ineligible subject matter] to allow the processes they describe to qualify as patent-eligible processes that *apply* [the subject matter]?").

[8] 134 S. Ct. 2347 (2014).

[9] *Id.* at 2358.

[10] *See infra* Part II.B. Whether interpreting § 101 in this manner is desirable from a policy perspective is a separate question outside the scope of this Note. For example, some have argued that the exceptions to patent eligibility under § 101 should be eliminated altogether in favor of more rigorous enforcement of the other provisions of the Patent Act. *See generally* Michael Risch, *Everything Is Patentable*, 75 TENN. L. REV. 591 (2008). This Note, however, builds on all Supreme Court precedent.

[11] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269 (Fed. Cir. 2013) (en banc), *aff'd*, 134 S. Ct. 2347 (2014).

[12] *See id.* at 1313 (Moore, J., dissenting in part).

[13] *See* Timothy B. Lee, *Software Patents Are Crumbling, Thanks to the Supreme Court*, VOX (Sept. 12, 2014, 3:50 PM), http://www.vox.com/2014/9/12/6138483/software-patents-are-crumbling-thanks-to-the-supreme-court.

[14] Bernard Chao, *Finding the Point of Novelty in Software Patents*, 28 BERKELEY TECH. L.J. 1217, 1222–23 (2013); Timothy B. Lee, *Software Patents Are a Disaster. The*

methods are simply not the type of innovation the Intellectual Property Clause of the United States Constitution[15] and the Patent Acts were meant to protect.[16] At the same time, other commentators have argued that many types of software innovation, especially innovation in the realm of algorithms themselves, both deserve patent protection and are appropriately encouraged through patents.[17] Whether the social benefits of software patents outweigh their social costs is an ongoing debate outside the scope of this Note. As things stand, software is not categorically ineligible for patent protection.[18] Thus, given the purpose of the patent-eligible subject matter exceptions endorsed by the Supreme Court, courts need a mechanism for distinguishing between software patents based on claim scope when deciding whether to invalidate a patent under § 101. Put differently, courts should be able to make a distinction between a pure business method patent that attempts to monopolize a broad abstract idea by only specifying that it is "implemented using a computer," and a software patent that narrowly claims what an inventor actually built, instead of categorically rendering all software patents running on a general-purpose computer invalid.

This Note argues that *Alice* and other Supreme Court precedent afford sufficient interpretive leeway to allow courts deciding questions of patent eligibility to make such a distinction. To accomplish this, additional claim limitations reciting specifics of how a piece of software accomplishes a patent-ineligible abstract idea should be "enough" to confer patentability,

*Courts Finally Did Something About It in 2014*, VOX (Dec. 31, 2014, 3:10 PM), http://www.vox.com/2014/12/31/7475317/software-patents-2014-review.

[15] U.S. CONST. art. I, § 8, cl. 8 ("To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries").

[16] Peter S. Menell, *Forty Years of Wondering in the Wilderness and No Closer to the Promised Land:* Bilski*'s Superficial Textualism and the Missed Opportunity to Return Patent Law to its Technology Mooring*, 63 STAN. L. REV. 1289, 1312–13 (2011).

[17] Donald S. Chisum, *The Patentability of Algorithms*, 47 U. PITT. L. REV. 959, 1013–17 (1986) (arguing in favor of making patent protection available to software algorithms). *But see* Allen Newell, *Response: The Models Are Broken, The Models Are Broken*, 47 U. PITT. L. REV. 1023, 1033–34 (1986) (arguing that software "may have a different character than the standard economic model of incentives that underlies the patent law"); Lee, *Software Patents Are a Disaster*, *supra* note 14.

[18] *See* Bilski v. Kappos, 130 S. Ct. 3218, 3226–27 (2010); *see also supra* Section I.C; *see also* Ca. Inst. of Tech. v. Hughes Commc'ns Inc., No. 2:13-cv-07245-MRP-JEM, 2014 WL 5661290, at *7–8 (C.D. Cal. Nov. 3, 2014) (arguing that "[t]he America Invents Act ('AIA') contemplates the existence of software patents explicitly" and that "the Supreme Court . . . implicitly endorsed the patentability of software").

even if the additional claim limitations are themselves patent-ineligible abstract ideas.

Parts I and II of this Note provide legal background; Part I surveys 35 U.S.C. § 101 jurisprudence, focusing on the cyclic rise and fall of the "enough" threshold marking the line between a patent-ineligible abstract idea and a patentable application thereof, and Part II discusses *Alice*. Part III explains the technical context of software and how it interacts with the patent system. Part IV analyzes how commentators, courts, and the United States Patent and Trademark Office ("PTO") have interpreted the *Alice* decision. Part IV also proposes an alternative interpretation that allows courts to decide the validity of software patents based on their actual preemptive effect and claim scope. Finally, Part V concludes.

## I. THE SEARCH FOR WHAT IS "ENOUGH": AN OVERVIEW OF § 101 JURISPRUDENCE

Section 101 dictates that "[w]hoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new useful improvement thereof, may obtain a patent."[19] Since software consists of a series of steps executed by a machine, claims directed at software routinely constitute a "process, machine, manufacture, or composition of matter" and are therefore at first glance patent-eligible under the statute.[20]

However, the Supreme Court has created judicial exceptions to these broad categories of patent-eligible subject matter.[21] As early as 1852, the Court in *Le Roy v. Tatham* held that "[a] principle, in the abstract, is a fundamental truth; an original cause, a motive; these cannot be patented, as no one can claim in either of them an exclusive right."[22] While the terminology used to describe and justify these exceptions has varied over time, today it is generally accepted that laws of nature, natural phenomena, and abstract ideas are not patent-eligible:

[19] 35 U.S.C. § 101 (2012).

[20] *See, e.g.*, CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1285–86 (Fed. Cir. 2013) (en banc) (Lourie, J., concurring), *aff'd*, 134 S. Ct. 2347 (2014); Ultramercial, Inc. v. Hulu, LLC, 722 F.3d 1335, 1349 (Fed. Cir. 2013), *vacated sub nom*, WildTangent, Inc. v. Ultramercial, LLC, 134 S. Ct. 2870 (2014), *aff'd*, 772 F.3d 709 (Fed. Cir. 2014); Digitech Image Techs., LLC v. Elec. For Imaging, Inc., 758 F.3d 1344, 1350 (Fed. Cir. 2014).

[21] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2354 (2014).

[22] 55 U.S. 156, 175 (1852).

> Laws of nature, natural phenomena, and abstract ideas are the basic tools of scientific and technological work. Monopolization of those tools through the grant of a patent might tend to impede innovation more than it would tend to promote it, thereby thwarting the primary object of the patent laws. [The Supreme Court has] repeatedly emphasized this concern that patent law not inhibit further discovery by improperly tying up the future use of these building blocks of human ingenuity.[23]

## A. THE EARLY, ELEVATED THRESHOLD OF "ENOUGH"—*BENSON* AND *FLOOK*

The first two Supreme Court patent eligibility decisions affecting software patents—*Gottschalk v. Benson* in 1972[24] and *Parker v. Flook* in 1978[25]—reflected the Court's underlying belief that software was categorically ineligible for patent protection.[26]

In *Benson*, the Court held that an algorithm, which today falls in the abstract idea category, is patent-ineligible.[27] The patent at issue claimed an algorithm for converting binary-coded decimal numbers, where each digit of the decimal number is expressed as a separate binary number (e.g., the number ten expressed as "0001 0000"), to a pure binary representation of that number (e.g., the number ten expressed as "1010").[28] This algorithm, the Court reasoned, was nothing more than a mathematical expression of a scientific truth, and as such was not patentable.[29] The Court's reasoning cast doubt on the availability of patent protection for software, since all software consists of retrieving numbers, storing numbers, and performing a limited set of simple arithmetic or logical functions on these numbers, regardless of how complex, sophisticated, or innovative the software is.[30]

[23] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2354 (citations and internal quotation marks omitted).

[24] 409 U.S. 63 (1972).

[25] 437 U.S. 584 (1978).

[26] *See Benson*, 409 U.S. at 71–72; *Flook*, 437 U.S. at 595–96.

[27] *Benson*, 409 U.S. at 67–68.

[28] *Id.* at 65–67.

[29] *Id.* at 67–68. Specifically, the Court considered the algorithm in *Benson* analogous to the mathematical expression at issue in *Mackay Radio and Telegraph Co. v. Radio Corp. of America*, 306 U.S. 86, 91–92 (1939)—a formula for calculating the angle at which a radio signal will be transmitted from an antenna based on the wave length of the signal and the length of the antenna wire.

[30] *See Benson*, 93 U.S. at 71–73; PATTERSON, *supra* note 1, at 78. This holding in *Benson* has been criticized by commentators. *See, e.g.*, Robert Merges, *Symposium: Go Ask* Alice—*What Can You Patent After* Alice v. CLS Bank*?*, SCOTUSBLOG (June 20, 2014, 12:04 PM), http://www.scotusblog.com/2014/06/symposium-go-ask-alice-what-can-you-patent-after-alice-v-cls-bank/:

Next, *Benson* held that the claims at issue did not include additional limitations that rendered the patent-ineligible algorithm patentable.[31] The Court noted that a claim is patent-eligible if it ties a patent-ineligible abstract idea to a particular machine, or uses it to change an article to a different state or thing.[32] However, the Court concluded that the claimed algorithm "has no substantial practical applications except in connection with a digital computer," and that the claim limitation of a computer was thus insufficient to render the claims patent-eligible.[33] While the Court noted that other types of limitations might also confer patent eligibility, *Benson* reflected a general pessimism on behalf of the Supreme Court regarding patent protection for software.[34]

> The Supreme Court [in *Benson* laid] down some notoriously ill-considered tracks in the domain of software patents. The essence of the decision was a ruling that the mathematical algorithm underlying the computer program was discovered rather than invented—it was a preexisting mathematical relationship which the 'inventor' merely harnessed for his patented invention. The resulting claim to the algorithm itself, was, the court decided, tantamount to a claim to a law of nature. This formulation of the claim misrepresents the nature of algorithms (which simply do not grow on trees), and has as a result spawned no end of confusion in the patent world.

After all, software algorithms do not occur naturally—instead, it is humans that select and combine mathematical operations when designing such software. *See id.* (algorithms "simply do not grow on trees"). Thus, the human contribution to software is no different than the human contribution to a steam engine—artificially selecting and combining patent-ineligible subject matter (metal and other compounds in the case of the steam engine, mathematical operations in the case of software) into something new and useful. *See* Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1294 (2012) ("The Court has recognized, however, that too broad an interpretation of this exclusionary principle could eviscerate patent law. For all inventions at some level embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract ideas."). However, the Supreme Court has continued to embrace the holding in *Benson.* Merges, *Go Ask* Alice, *supra* ("But now the Supreme Court has gone and assimilated the *Benson* holding into the new *Bilski-Mayo-Alice* framework, in a way that will surely bring future headaches."); *see also infra* Section IV.A.1. Thus, this Note accepts that algorithms are patent-ineligible as a given and incorporates the holding in *Benson* into the framework proposed in Part IV.

[31] *Benson*, 409 U.S. at 70–71. In later cases, the Court phrased this type of inquiry as deciding whether "the patent claims add *enough* to [patent-ineligible subject matter] to allow the processes they describe to qualify as patent-eligible processes that *apply* the" patent-ineligible subject matter. *See Mayo*, 132 S. Ct. at 1297.

[32] *Benson*, 409 U.S. at 70–71.

[33] *Id.* at 71–72.

[34] *Id.* at 71–73.

The threshold of "enough" set forth in *Benson* was further heightened five years later in *Flook*.[35] Driven by concerns that a "competent draftsman" should not be able to transform a patent-ineligible abstract idea into a patent-eligible application by appending "conventional or obvious" "post-solution activity" (i.e., additional physical steps taken after a mathematical expression is solved),[36] the majority held that a claim reciting an abstract idea must also include an "inventive concept" to be patent-eligible.[37] Specifically, when analyzing such a claim, a court should assume that the abstract is already present in the prior art and then determine whether the remainder of the claim is new and useful—i.e., whether the remainder of the claim constitutes a "patentable invention."[38] Under this "inventive concept" standard, if an inventor's contribution to human knowledge, sometimes referred to as the point of novelty, [39] lies in the abstract idea itself, it is not patent-eligible.[40] Thus, under *Flook*, a new algorithm (i.e., an abstract idea), without accompanying new and useful hardware, is not eligible for patent protection, regardless of how revolutionary the algorithm may be, how narrowly it is claimed, what real-world applications it may have, or how much it may improve existing physical processes.[41]

The combination of *Benson* and *Flook* virtually eliminated patent protection for software—*Benson* labeled algorithms, and thus by extension all software, as patent-ineligible abstract ideas, and *Flook* by indicating that, as long as an invention's sole point-of-novelty lay in the software, no additional limitation could be "enough" to confer patent eligibility.

[35] *Compare id.* at 71–72 (finding that the claimed digital computer did not confer patent eligibility onto the abstract idea because the digital computer did not prevent the claim from "*wholly* pre-empt[ing]" the abstract idea (emphasis added)) *with* Parker v. Flook, 437 U.S. 584, 595–96 (1978) (finding that the claimed chemical process did not confer patent eligibility onto the abstract idea because, absent the abstract idea, it was not patentable).

[36] *Flook*, 437 U.S. at 590.

[37] *Id.* at 594.

[38] *Id.*

[39] *See, e.g.*, Diamond v. Diehr, 450 U.S. 175, 200 n.15 (1981) (Stevens, J., dissenting).

[40] Chao, *Finding the Point of Novelty*, *supra* note 14, at 1241.

[41] *Id.* at 1241–42; *see also* Mark A. Lemley, *Point of Novelty*, 105 NW. U.L. REV. 1253, 1278 (2011).

## B. LOWERING THE PATENT ELIGIBILITY THRESHOLD—FROM *DIEHR* TO *STATE STREET BANK*

Only three years after *Flook*, in 1981 the Supreme Court reversed direction in *Diamond v. Diehr*, and lowered the "enough" threshold sufficiently to make patent protection available to software innovation.[42] Even though *Diehr* did not explicitly overrule *Flook*, the two cases are difficult, if not impossible, to harmonize.[43]

The *Diehr* majority viewed the claims as directed to a process for transforming rubber, and held that the fact that one of the steps of this otherwise patentable process involved a mathematical expression did not deprive the whole process of patent eligibility.[44] While *Diehr* acknowledged that limiting an abstract idea "to a particular technological environment" or including "insignificant post-solution activity" cannot confer patent eligibility,[45] the Court also warned that a claim cannot be dissected and must be considered as a whole.[46] The majority in *Diehr* thereby redefined what constitutes "enough" to confer eligibility on a patent-ineligible abstract idea—any limitation that ensures that the claimed invention, "when considered as a whole, is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article to a different state or thing)."[47]

The Court of Customs and Patent Appeals ("C.C.P.A."), the predecessor to the Federal Circuit, implemented the holding in *Diehr* as

[42] *See* 450 U.S. at 192. While the same Justices were on the Court in both cases, Justice White and Justice Powell switched sides—excluding them, the dissenters in *Diehr* formed the majority in *Flook*, and vice versa. *Id.* at 177, 193.

[43] Chao, *Finding the Point of Novelty*, *supra* note 14, at 1235.

[44] *Diehr*, 450 U.S. at 191–92.

[45] *Id.* at 191–92.

[46] *Id.* at 188.

[47] *Id.* at 192. The dissenting opinion in *Diehr*, written by Justice Stevens who wrote for the majority in *Flook*, would have maintained the standard set forth in *Flook*—if the entire subject matter an inventor claims to be novel (i.e., the inventor's contribution or the point the novelty) falls within a patent-ineligible exception to § 101, the claim as a whole is patent-ineligible, regardless of what additional limitations are included. *See id.* at 211–12 (Stevens, J., dissenting). In fact, the dissent would have preferred to make a unequivocal and sweeping proclamation: (1) "no program-related invention is a patentable process under § 101 unless it makes a contribution to the art that is not dependent entirely on the utilization of a computer," and (2) "the term 'algorithm' as used in [*Diehr*], as in *Benson* and *Flook*, is synonymous with the term 'computer program.'" *Id.* at 219. In essence, the dissent argued that software innovation is categorically patent-ineligible, and that only an additional, separate, novel, and non-obvious hardware element would be "enough" to confer patent eligibility.

the *Freeman-Walter-Abele* test.[48] Under this test, a claim had to apply an abstract idea to a physical element or as part of a physical process in order to be patent-eligible.[49] For example, in *In re Abele*, the C.C.P.A. held that a claim directed to calculating numbers based on collected data and displaying these numbers was patent-ineligible.[50] However, the C.C.P.A. also held, in the same case, that a claim which further required the collected data to be X-ray attenuation data from a computed axial tomography ("CAT") scan, was patent-eligible, since the CAT scan was a physical process and part of the claim.[51]

Over time, the C.C.P.A., and later the Federal Circuit, further lowered the "enough" threshold. Initially, the court liberally construed what constitutes a physical process. In *Arrhythmia Research Technology Inc. v. Corazonix Corp.*, the Federal Circuit determined that transforming one electrical signal into another within a computer was a physical process and therefore "enough" to confer patent eligibility. [52] Once this reasoning took hold, the *Freeman-Walter-Abele* test met its end in *In re Alappat*, where the Federal Circuit held that transforming one set of data into another set of data was "enough" for patent eligibility, since the transformation produced "a useful, concrete, and tangible result."[53] Finally, the lowest threshold of what additional limitations are "enough" to confer patent eligibility was set in *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*[54] There, the Federal Circuit formally overruled the *Freeman-Walter-Abele* test and held that any claim incorporating an abstract idea that produces "a useful, concrete, and tangible result" is patent-eligible.[55]

Thus the barrier to patenting software created by *Benson* and *Flook* was completely eroded in subsequent decisions. By the start of the twenty-first century, software was eligible for patent protection.

[48] Arrhythmia Research Tech., Inc. v. Corazonix Corp., 958 F.2d 1053, 1057–58 (Fed. Cir. 1992).

[49] *Id.* at 1058

[50] 684 F.2d 902, 908 (C.C.P.A. 1982).

[51] *Id.* at 908–09.

[52] 958 F.2d at 1059.

[53] 33 F.3d 1526, 1544 (Fed. Cir. 1994).

[54] 149 F.3d 1368, 1373–75 (Fed. Cir. 1998).

[55] *Id.*

### C. THE PATENT ELIGIBLITY THRESHOLD RISES AGAIN—*BILSKI* AND *MAYO*

Uncertainty regarding software's patent eligibility returned in the wake of the Supreme Court's 2010 *Bilski v. Kappos*[56] and 2012 *Mayo Collaborative Services v. Prometheus Laboratories, Inc.* decisions.[57] Unlike *Flook* and *Benson*, neither of these decisions ever questioned whether patent protection was available for software.[58] At the same time, the Court noticeably raised the "enough" threshold for patent eligibility, so that § 101 again became a substantive hurdle for software.[59] Thus, while neither *Bilski* nor *Mayo* categorically excluded software innovation from the patent system, the two cases in effect rendered the validity of most software patents uncertain.

The tide started turning against software patents in 2008 when the Federal Circuit held in *In re Bilski* that the appropriate § 101 inquiry is whether a claim meets the "machine-or-transformation test."[60] Under this new test, a claim incorporating an abstract idea includes "enough" to transform it into a patent-eligible application *if and only if* the abstract idea is either (1) "tied to a particular machine or apparatus," or (2) "transforms a particular article into a different state or thing."[61] The court did not resolve how high a threshold this was or how difficult it would be for software patents to meet, since the court declined to decide whether a general-purpose computer constitutes "a particular machine or apparatus."[62] Moreover, the court noted that transforming data *representing* a physical object and then displaying this data was sufficient to meet the transformation prong of the test.[63]

The Supreme Court responded with its 2010 *Bilski* decision. The Court first addressed the Federal Circuit's new machine-or-transformation test, concluding that, while it was "an important and useful clue," it could not be the sole test for determining whether an invention is patent-eligible:

[56] 130 S. Ct. 3218 (2010).

[57] 132 S. Ct. 1289 (2012).

[58] *See Bilski*, 130 S. Ct. at 3226–27 (indicating that the Federal Circuit's machine-or-transformation test cannot be the sole test for patent eligibility, since it brought uncertainty regarding the availability of patent protection for software innovation).

[59] *See Mayo*, 132 S. Ct. at 1297–98 (implying that only activity that was not "well-understood, routine, [or] conventional" was "enough" to confer patent eligibility).

[60] 545 F.3d 943, 954 (Fed. Cir. 2008) (en banc), *aff'd*, 130 S. Ct. 3218 (2010).

[61] *Id.*

[62] *Id.* at 962.

[63] *Id.* at 963.

> The machine-or-transformation test may well provide a sufficient basis for evaluating processes similar to those in the Industrial Age—for example, inventions grounded in a physical or other tangible form. But there are reasons to doubt whether the test should be the sole criterion for determining the patentability of inventions in the Information Age. As numerous *amicus* briefs argue, the machine-or-transformation test would create uncertainty as to the patentability of software, advanced diagnostic medicine techniques, and inventions based on linear programming, data compression, and the manipulation of digital signals.[64]

Next, the Court held that business methods, just like other types of innovation, are not categorically barred from patent protection, unless the particular business method at issue also happens to be an abstract idea.[65]

Importantly, the Court reinvigorated *Flook* and attempted to harmonize it with *Diehr*. The Court described the *Flook* claim as an attempt to patent a mathematical algorithm, adding only insignificant post-solution activity and limiting it to a field of use, and the *Diehr* claim as being directed to an industrial process that happens to include a mathematical formula.[66] In so doing, the Court did not explain why the claims in *Flook* and *Diehr* warranted such drastically different descriptions,[67] when, superficially at least, they appeared to be very similar.[68]

Finally, the Court concluded that the *Bilski* claims were not patent-eligible because the concept of hedging recited therein was an unpatentable abstract idea, and allowing the claims would preempt and effectively grant a monopoly over this abstract idea.[69] In reaching this conclusion, the Court did not elaborate on, or even hint at what additional limitations would be "enough" to render the claim patent-eligible.

While the Supreme Court's decision in *Bilski* was superficially more permissive of software patents than the corresponding Federal Circuit

[64] Bilski v. Kappos, 130 S. Ct. 3218, 3226–27 (2010).

[65] *Id.* at 3228–29. The concurrence in *Bilski*, following an extensive analysis of the history of U.S. patent law, concluded that a method of conducting business is not a "process" under § 101. *Id.* at 3239–50 (Stevens, J., concurring). Based on this interpretation, the concurrence argued that the claims at issue are invalid not for falling into the abstract idea exception to § 101, but for not being directed to any one of the categories of patent-eligible subject matter in the first place. *Id.* at 3257.

[66] *Id.* at 3230 (majority opinion).

[67] *Id.*

[68] Chao, *Finding the Point of Novelty*, *supra* note 14, at 1235.

[69] *Bilski*, 130 S. Ct. at 3231.

decision, it also endorsed the prior Supreme Court opinions most detrimental to software patents—*Benson* and *Flook*.

The Supreme Court's next § 101 decision, *Mayo*, further endorsed *Benson* and *Flook*. There, the Court held that a claim must do more than recite a patent-ineligible law of nature and instruct the relevant audience to apply it—it must also include an "inventive concept," a term borrowed from *Flook*.[70] However, the Court did not explain whether this term had the same meaning as in *Flook*.[71] Nor did the Court assume that the claimed patent-ineligible subject was in the prior art and then evaluate the patentability of the claim, as it did in *Flook*.[72] Instead, the Court held that obvious, well-understood, routine, and conventional activity already engaged in by the scientific community is not "enough" to confer patent eligibility.[73] The Court recognized that this inquiry may sometimes, but not always, involve an "overlap" between "the § 101 patent eligibility inquiry and . . . the § 102 novelty inquiry."[74] The Court also continued to maintain that *Flook* and *Diehr* are non-conflicting valid precedent, relying on a similar description of these two cases as in *Bilski*.[75]

After *Bilski* and *Mayo*, the Federal Circuit issued a series of inconsistent decisions regarding the patent eligibility of claims directed to computer-implemented software or algorithms (i.e., abstract ideas). One faction of the court argued that a computer must "play a significant part in" or "be integral to" the claimed invention, "facilitating the process in a way that a person making calculations or computations could not" for the limitation to be "enough" to confer patent eligibility to an otherwise patent-ineligible abstract idea.[76] The other faction argued that unless a claim is so lacking in additional limitation that it is "manifestly evident that [it] is directed to a patent-ineligible abstract idea," a claim incorporating an abstract idea is patent-eligible.[77] During this time, the

[70] Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1294 (2012).

[71] *Id.*

[72] *Id.* at 1294–99.

[73] *Id.* at 1298.

[74] *Id.* at 1304.

[75] *Id.* at 1299.

[76] Bancorp Servs. L.L.C. v. Sun Life Ins. Co. of Can. (U.S.), 687 F.3d 1266, 1278 (Fed. Cir. 2012).

[77] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 685 F.3d 1341, 1352 (Fed. Cir. 2012), *vacated*, 717 F.3d 1269 (Fed. Cir. 2013) (en banc), *aff'd*, 134 S. Ct. 2347 (2014).

outcome of any particular case depended more on which judges heard the case, rather than the actual merits and the claim language.[78]

Thus, *Bilski* and *Mayo* cast uncertainty on when, if ever, software executed by a physical computer is patent-eligible and, more generally, what limitations are "enough" to confer patent eligibility to an abstract idea. Commentators were eagerly awaiting a cause that would resolve the uncertainty in § 101 jurisprudence as applied to software.[79]

## II. THE SUPREME COURT'S *ALICE V. CLS BANK* DECISION

Part II discusses *Alice*. Section II.A provides a history of the case, including the fundamental disagreement at the Federal Circuit level that lead to a fractured en banc decision. Section II.B then discusses the Supreme Court's analysis and holding in *Alice*.[80]

[78] Bernard Chao, *Interpreting* CLS Bank Int'l v. Alice, PATENTLY-O (Sept. 3, 2013), http://patentlyo.com/patent/2013/09/interpreting-cls-bank-intl-v-alice.html (observing that, even after the *CLS Bank* en banc decision, "success of any appeal to the Federal Circuit may simply be panel dependent"). The Federal Circuit issued six decisions between March 20, 2012, when the Supreme Court decided *Mayo*, and June 19, 2014, when the Supreme Court decided *Alice*, where at least one judge of the Federal Circuit that took part in *CLS Bank* also expressed an opinion regarding the patent eligibility of another software patent. CLS Bank Int'l v. Alice Corp. Pty. Ltd., 685 F.3d 1341 (Fed. Cir. 2012); Bancorp Servs. L.L.C. v. Sun Life Ins. Co. of Can. (U.S.), 687 F.3d 1266 (Fed. Cir. 2012); Ultramercial, Inc. v. Hulu, LLC, 722 F.3d 1335 (Fed. Cir. 2013); Accenture Global Servs., GmbH v. Guidewire Software, Inc., 728 F.3d 1336 (Fed. Cir. 2013); SmartGene Inc. v. Advanced Biological Labs., SA, 555 F. App'x 950 (Fed. Cir. 2014); Cyberfone Sys., LLC v. CNN Interactive Grp., Inc., 558 F. App'x 988 (Fed. Cir. 2014). In these decisions, the judges that found the system claims at issue in *CLS Bank* to be patent-eligible also found these other software patents patent-eligible five out of five times. *CLS Bank Int'l*, 685 F.3d at 1343; *Ultramercial, Inc.*, 722 F.3d at 1337; *Accenture Global Servs., GmbH*, 728 F.3d at 1347–48 (Rader, C.J., dissenting). On the other hand, the judges that held that all claims in *CLS Bank* were invalid also found these other software patents patent-ineligible eleven out of twelve times. *CLS Bank Int'l*, 685 F.3d at 1356 (Prost, J., dissenting); *Bancorp Servs. L.L.C.*, 687 F.3d at 1269; *Ultramercial, Inc.*, 722 F.3d at 1355 (Lourie, J., concurring); *Accenture Global Servs., GmbH*, 728 F.3d at 1337; *SmartGene Inc.*, 555 F. App'x at 951; *Cyberfone Sys., LLC*, 558 F. App'x at 989. The only time any Federal Circuit judge broke rank was when Judge Lourie found the claims at issue in *Ultramercial, Inc. v. Hulu, LLC* valid under § 101 and thought they ought to survive a motion to dismiss. *Ultramercial, Inc.*, 722 F.3d at 1355 (Lourie, J., concurring).

[79] *See, e.g.*, Chao, *Interpreting* CLS Bank Int'l v. Alice, *supra* note 78 ("Hopefully, the Supreme Court will resolve" the issue that, after *CLS Bank Int'l*, "the success of any appeal to the Federal Circuit may simply be panel dependent."); Guy Gosnell & Jim Carroll, CLS Bank *Ruling Won't Have Much Impact On Practitioners*, LAW360 (June 17, 2013, 12:31 PM), http://www.law360.com/articles/448529/cls-bank-ruling-won-t-have-much-impact-on-practitioner.

[80] *See also supra* Part IV (discussing possible interpretations and effects of *Alice*).

### A. PROCEDURAL HISTORY

On May 24, 2007, CLS Bank International sued Alice Corp. Pty. Ltd. seeking declaratory judgment of non-infringement, invalidity, and unenforceability of U.S. Patent No. 5,970,479 ("the '479 patent"), U.S. Patent No. 6,912,510 ("the '510 patent"), U.S. Patent No. 7,149,720 ("the '720 patent"), and U.S. Patent No. 7,725,375 ("the '375 patent").[81] The asserted claims covered a form of escrow that allows parties to enter into a contract scheduled to mature at a later time.[82] For example, two individuals may agree in the morning to exchange currencies, but, to avoid allowing either party to gain a benefit by exactly timing its acceptance based on fluctuations in the exchange rate, the two individuals might agree that the exchange rate at the end of the day is controlling.[83] When forming such a contract, a mechanism is necessary to ensure that each party is still able to pay its side of the bargain at the end of the day.[84] The claimed invention solves this issue by having a trusted third-party keep track, via a so-called shadow credit and/or debit record, of a contracting party's financial transactions between the time a contract is initially formed (e.g., in the morning, when the two individuals agree to exchange currencies) and when it matures (e.g., at the end of the day, when the controlling exchange rate is set).[85] The third-party blocks any financial transactions that would prevent a contracting party from performing its side of the contract.[86] Once the contract matures, the third-party instructs an exchange institution (e.g., the contracting party's bank) to perform non-blocked financial transactions.[87] The claims asserted in this case included method, computer system, and computer-readable medium claims directed to this mechanism.[88]

At the district court, CLS Bank moved for summary judgment after initial discovery, alleging that all asserted claims were directed to patent-ineligible subject matter.[89] Since claim construction had not yet taken place, the district court presumed "that the terms 'shadow' credit and/or

[81] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 768 F. Supp. 2d 221, 223, 228–29 (D.D.C. 2011), *aff'd*, 717 F.3d 1269 (Fed. Cir. 2013) (en banc), *aff'd*, 134 S. Ct. 2347 (2014).

[82] *Id.* at 224–225.

[83] *See id.*

[84] *See id.*

[85] *See id.*

[86] *See id.*

[87] *See id.*

[88] *Id.* at 224–28

[89] *Id.* at 228.

debit record and 'transaction' in [the method claims] recite electronic implementation and a computer or an analogous electronic device."[90] The court also defined the abstract idea as "[u]sing an intermediary, which may independently maintain records or accounts on the parties to ensure each party has sufficient value or worth to complete a proposed exchange, as a way to guarantee that a transaction is ultimately honored by all parties, thereby minimizing risk."[91] The court then found all asserted claims invalid under § 101 for preempting this abstract idea.[92]

On appeal, a three-judge panel of the Federal Circuit reversed the district court's decision, holding that it was not manifestly evident that the asserted claims were directed to patent-ineligible subject matter.[93]

The Federal Circuit then granted CLS Bank's petition for an en banc rehearing to decide (a) what test courts should adopt to determine whether a computer-implemented invention is a patent-ineligible abstract idea, including whether the recitation of a computer is "enough" to confer patent eligibility, and (b) whether the manner in which an invention is claimed (i.e., as a method, system, or storage medium) affects this determination.[94] Instead of definitively answering either of these questions, the en banc panel issued a *per curiam* decision with five opinions on the merits proposing "at least three incompatible standards, devoid of consensus,"[95] and "Additional reflections" by Chief Judge Rader lamenting

[90] *Id.* at 237.

[91] *Id.* at 244.

[92] *Id.* at 245–53, 255.

[93] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 685 F.3d 1341, 1356 (Fed. Cir. 2012), *vacated*, 717 F.3d 1269 (Fed. Cir. 2013) (en banc), *aff'd*, 134 S. Ct. 2347 (2014).

[94] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 484 F. App'x 559 (Fed. Cir. 2012) (order granting rehearing en banc).

[95] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1321 (Fed. Cir. 2013) (en banc) (Newman, J., concurring in part and dissenting in part), *aff'd*, 134 S. Ct. 2347 (2014). Judge Lourie wrote the first opinion, which was joined by Judges Dyk, Prost, Reyna, and Wallach. *Id.* at 1273 (Lourie, J., concurring). Chief Judge Rader wrote a second opinion, which was joined by Judge Moore and partially by Judges Linn and O'Malley. *Id.* at 1313 (Rader, C.J., concurring in part and dissenting in part). Judges Linn and O'Malley wrote a third opinion agreeing with Chief Judge Rader's analysis, but disagreeing over how the method claims should be construed, believing them to be equivalent to the system claims. *Id.* at 1327 (Linn, J., and O'Malley, J., dissenting). Judge Moore also wrote a separate fourth opinion joined by Chief Judge Rader and Judges Linn and O'Malley (i.e., the judges who also joined Chief Judge Rader's opinion), where he warned that endorsing Judge Lourie's theory of § 101 would lead to "the death of hundreds of thousands of patents." *Id.* at 1313 (Moore, J., dissenting in part). Judge Newman issued a final fifth opinion, arguing for the elimination of the exceptions to patent-eligible subject matter. *Id.* at 1321 (Newman, J., concurring in part and dissenting in part).

the state of § 101 jurisprudence.[96] Chief Judge Rader later went on to describe the *CLS Bank* en banc decision as "the greatest failure in [his] judicial career."[97]

The two main opinions, one written by Judge Lourie and the other by Chief Judge Rader, reflect two diverging perspectives on the purpose of the patent subject matter exceptions.[98] Even with this difference in perspective, Judge Lourie and Chief Judge Rader agreed on how to interpret prior Supreme Court precedent and the method and storage medium claims at issue.[99] However their fundamental disagreement on the purpose of the exceptions to patent-eligible subject matter, when coupled with the nature of computers, caused the two judges to diverge on the system claims.[100]

Judge Lourie's and Chief Judge Rader's opinions each reflect one of two views on the purpose of the exceptions to patent-eligible subject matter—as a substantive limitation on patent scope to preclude issuance of overly broad patents,[101] or as a coarse filter to preclude patents on certain types of discovery that would impede more than encourage progress.[102]

[96] *Id.* at 1333–35 (Rader, C.J., additional reflections).

[97] Tony Dutra, *Rader Regrets* CLS Bank *Impasse, Comments on Latest Patent Reform Bill*, BLOOMBERG BNA (Oct. 29, 2013), http://www.bna.com/rader-regrets-cls-n17179879684.

[98] *Compare CLS Bank Int'l*, 717 F.3d at 1280–82 (Lourie, J., concurring) (arguing that a claim is patent-eligible if it includes "substantive limitations" so that the claim does not in practice cover an abstract idea) *with id.* at 1299–1300 (Rader, C.J., concurring in part and dissenting in part) (arguing that a claim is patent-eligible if it is "tie[d] to a concrete reality").

[99] *Compare id.* at 1280–89 (Lourie, J., concurring) (finding the method claims patent-ineligible) *with id.* at 1299–1305, 1311–13 (Rader, C.J., concurring in part and dissenting in part) (finding the method claims patent-ineligible).

[100] *Compare id.* at 1289–92 (Lourie, J., concurring) (finding the system claims patent-ineligible) *with id.* at 1306–11 (Rader, C.J., concurring in part and dissenting in part) (finding the system claims patent-eligible). Additionally, a majority of judges agreed that the presumption of validity under § 282 also applies to challenges under § 101. *Id.* at 1284 (Lourie, J., concurring), 1304–05 (Rader, C.J., concurring in part and dissenting in part). Accordingly, an alleged infringer must prove by clear and convincing evidence that a patent is directed to a patent-ineligible abstract idea to invalidate it. Moreover, Judge Lourie's conclusion that district courts do not have to address a § 101 challenge to an issued patent as a threshold matter and before addressing other theories of invalidity was not contested by any of the other judges. *Id.* at 1284 (Lourie, J., concurring)

[101] *See, e.g.*, Robert D. Swanson, Note, *Section 101 and Computer-Implemented Inventions*, 16 STAN. TECH. L. REV. 161, 174 (2012); Mark A. Lemley et al., *Life After* Bilski, 63 STAN. L. REV. 1315, 1317 (2011).

[102] *See, e.g.*, Joshua A. Kresh, Note, *Patent eligibility After* Mayo*: How Did We Get Here and Where Do We Go?*, 22 FED. CIR. B.J. 521, 521 (2013); Swanson, *supra* note 101, at 174.

Judge Lourie subscribed to the substantive limitation theory of § 101. According to his opinion, if a claim incorporates an abstract idea, it must also include "additional substantive limitations that narrow, confine, or otherwise tie down the claim so that, in practical terms, it does not cover the full abstract idea itself" and thus "preempt [one of] the fundamental tools of discovery."[103] In contrast, Chief Judge Rader sided with the coarse filter theory. According to his opinion, the relevant inquiry under § 101 is whether "the claim contains limitations that meaningfully tie that idea *to a concrete reality* or actual application of that idea."[104]

Even with their disagreement on the purpose of the patent-eligible subject matter exceptions, Judge Lourie and Chief Judge Rader still shared a similar view of precedent and on many of the claims in the case. Thus, while the two judges disagreed over how to characterize the "inventive concept" referenced in *Mayo*, they agreed that "inventiveness" did not imply a requirement of novelty or non-obviousness.[105] Additionally, the two judges generally agreed on what types of limitations are *not* "enough"

[103] *CLS Bank Int'l*, 717 F.3d at 1280–82 (Lourie, J., concurring).

[104] *Id.* at 1299–1300 (Rader, C.J., concurring in part and dissenting in part) (emphasis added).

[105] Chief Judge Rader argued that the Supreme Court's reference to "inventive concept" should not be read "to instill an 'inventiveness' or 'ingenuity' component into the [§ 101] inquiry," but instead as "shorthand for [the Supreme Court's] inquiry into whether implementing the abstract idea in the context of the claimed invention inherently requires the recited steps." *Id.* at 1302–03 (Rader, C.J., concurring in part and dissenting in part). Judge Lourie, on the other hand, argued that the "'inventive concept' . . . refers to a genuine human contribution to the claimed subject matter" and "a product of human ingenuity." *Id.* at 1283 (Lourie, J., concurring) (quoting *Diamond v. Chakrabarty*, 447 U.S. 303, 307 (1980) (internal quotation marks omitted). However, even under Judge Lourie's interpretation, an "inventive concept" does not require "'inventiveness' in the same sense as that term more commonly applies to two of the statutory requirements for patentability, i.e., novelty and nonobviousness" and that the Supreme Court's references to "routine" or "conventional" steps is to be understood as a question of whether "a claim . . . effectively covers the natural law or abstract idea itself." *Id.* at 1282–84. Thus, while the two judges used different language, both agreed that the search for an "inventive concept" does not overlap with §§ 102 and 103 and is instead focused on whether the additional limitations actually limit the claim so that it is not coextensive with the abstract idea. Accordingly, Chief Judge Rader's criticism of Judge Lourie's definition of "inventive concept" as "imbu[ing] it with a life that is neither consistent with the Patent Act's description of Section 101 nor with the totality of Supreme Court precedent" and "inject[ing] an 'ingenuity' requirement" is more a reflection of the deeply divided nature of the court than of any practical differences between the two definitions of "inventive concept." *See id.* at 1303 n.5 (Rader, C.J., concurring in part and dissenting in part).

to confer patent eligibility.[106] As a result, the two judges agreed that the method and storage medium claims at issue were patent-ineligible, although their respective opinions use different language.[107]

However, Judge Lourie and Chief Judge Rader passionately disagreed on how to treat the concrete recitations of computer hardware found in the system claims. This divergence stems from the very nature of computers. On one hand, computers are physical appliances that can be put on a desk and thus meet Chief Judge Rader's coarse filter of being grounded in a concrete and physical reality.[108] According to Chief Judge Rader's view, a concrete recitation of computer hardware is "enough" to confer patent eligibility if "the computer plays a meaningful role in the performance of the claimed invention."[109] Based on this premise, Chief

[106] According to Judge Lourie, "a trivial appendix to the underlying abstract idea," limitations that "in practice fail to narrow the claim relative to the fundamental principle therein," "token or trivial limitations," "vague limitations cast in 'highly general language,'" "bare field-of-use limitations," and limitations that "are so insignificant, conventional, or routine as to yield a claim that effectively cover the natural law or abstract idea itself," do not render a claim otherwise directed to a patent-ineligible abstract idea patentable. *Id.* at 1283–84 (Lourie, J., concurring). Similarly, Chief Judge Rader noted that a claim is not meaningfully limited if it "merely describes an abstract idea or simply adds 'apply it,'" "contains only insignificant or token pre- or post-solution activity—such as identifying a relevant audience, a category of use, field of use, or technological environment," or "provide[s] no real direction, cover[s] all possible ways to achieve the provided result, or [is] overly-generalized." *Id.* at 1300–01 (Rader, C.J., concurring in part and dissenting in part).

[107] *Id.* at 1285–89 (Lourie, J., concurring) ("[T]here is nothing in the asserted method claims that represents '*significantly more*' than the underlying abstract idea for purposes of §101. . . . [U]pholding Alice's claims . . . 'would pre-empt use of this approach in all fields, and would effectively grant a monopoly over an abstract idea.'"); *id.* at 1311–13 (Rader, C.J., concurring in part and dissenting in part) ("Viewed individually, the recited elements only recite the steps *inherent* in [the abstract] concept (stated at a high level of generality and implement those steps according to methods long used in escrows according to the record in this case." (emphasis added)).

[108] *See id.* at 1299–1300 (Rader, C.J., concurring in part and dissenting in part) ("A claim may be premised on an abstract idea—the question for patent eligibility is whether the claim contains limitations that meaningfully tie that idea *to a concrete reality* or actual application of that idea." (emphasis added)); *id.* at 1320 (Moore, J., dissenting in part) ("Looking at these hardware and software elements, it is impossible to conclude that this claim is merely an abstract idea. It is a pure system claim, directed to a specific machine configured to perform certain functions. Indeed, the computer covered by this claim is a tangible item that you could pick up and put on your desk.").

[109] *Id.* at 1302 (Rader, C.J., concurring in part and dissenting in part). The requirement of a meaningful role ensures that the recitation of computer hardware is not merely tangential to the claimed invention and thus "insignificant or token pre- or post-solution activity" insufficient to confer patent eligibility. *See id.* at 1300–01.

Judge Rader found the system claims at issue patent-eligible.[110] On the other hand, all computers include generic hardware elements and perform the same basic functions.[111] Accordingly, the recitation of generic computer hardware is in practice, due to the universal presence of computers in modern life, not a substantive limitation on claim scope, as required for patent eligibility under Judge Lourie's substantive limitation analysis.[112] Thus, according to Judge Lourie, "[u]nless the claims require a computer to perform operations that are not merely accelerated calculations, a computer does not itself confer patent eligibility."[113] In view of this requirement, Judge Lourie found the system claims at issue patent-ineligible.[114]

Therefore, while Judge Lourie and Chief Judge Rader converged with regards to the method and storage medium claims, their disagreement on the purpose of the subject matter eligibility exceptions to patent-eligible subject matter caused the two judges to diverge with regards to the system claims. The result was a "hopelessly fractured" Federal Circuit.[115] The judges that agreed with Chief Judge Rader also joined an opinion written by Judge Moore arguing that Judge Lourie's approach would cause "the death of hundreds of thousands of patents, including all business method, financial system, and software patents as well as many computer implemented and telecommunications patents."[116] These judges accused Judge Lourie of being plain "wrong" and of "trampl[ing] upon a mountain of precedent."[117] Moreover, Chief Judge Rader's "Additional reflections" lamented recent Supreme Court precedent regarding § 101, calling it "a good mystery" containing "subjective and empty words" that left the Federal Circuit "with little, if any, agreement . . . even though [§ 101] has not changed a syllable."[118] Given this chaos, Supreme Court intervention

[110] *Id.* at 1306–11.

[111] *See* Merges, *Go Ask* Alice, *supra* note 30 ("novel hardware elements are simply not a prominent feature of many software patents").

[112] *See CLS Bank Int'l*, 717 F.3d at 1286 (Lourie, J., concurring).

[113] *Id.* Chief Judge Rader heavily criticized this requirement, arguing that it meant that "computer implementation could never produce patent eligibility," since "[e]verything done by a computer can be done by a human." *Id.* at 1306 n.7 (Rader, C.J., concurring in part and dissenting in part).

[114] *Id.* at 1289–92 (Lourie, J., concurring).

[115] Gosnell, *supra* note 79.

[116] *CLS Bank Int'l*, 717 F.3d at 1313 (Moore, J., dissenting in part).

[117] *Id.* at 1313–14.

[118] *Id.* at 1335 (Rader, C.J., additional reflections).

was almost inevitable,[119] and, as anticipated, the Court granted the petition for writ of certiorari. [120]

### B. THE SUPREME COURT DECISION

The Court's skepticism regarding the patentability of the claims at issue in *Alice* was foreshadowed during oral arguments. Justice Breyer questioned whether the claims amounted to anything more than an accounting system in use since the days of King Tut, except that the system was implemented using a computer instead of a person holding an abacus.[121] Justice Kennedy noted that the innovative aspect of the claimed invention did not lie in implementing the abstract idea at issue on a computer, but in the abstract idea itself.[122] Justice Kennedy thus questioned whether, given the abstract idea, "any computer group of people sitting around a coffee shop in Silicon Valley" or "any second-year college class in engineering" could program the software necessary to implement the claimed invention.[123]

So, it came as no surprise when the Supreme Court unanimously decided that all of the claims at issue in *Alice* were patent-ineligible.[124] The holding of the case is a straightforward application of precedent—since an abstract idea does not become patent-eligible by merely adding the words "apply it"[125] or by limiting it "to a particular technological environment,"[126] "[s]tating an abstract idea while adding the words 'apply it with a computer' simply combines those two steps" and does not confer patent eligibility.[127] The Supreme Court then decided that the claims at issue, including all of the method, system, and storage medium claims, do no more than "simply instruct the practitioner to implement the abstract idea

[119] *See* Chao, *Interpreting* CLS Bank Int'l v. Alice, *supra* note 78; Gosnell, *supra* note 79.

[120] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 734, 735 (2013) (granting petition for writ of certiorari).

[121] Transcript of Oral Argument at 6, Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014) (No. 13-298).

[122] *Id.* at 5.

[123] *Id.* at 5, 12.

[124] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2352. All justices joined the majority opinion. *Id.* at 2360–61 (Sotomayor, J., concurring). However, Justices Sotomayor, Ginsburg, and Breyer maintained the reasoning found in the concurring opinion of Justice Stevens in *Bilski*. *Id.* at 2360–61. According to these three Justices, the claims at issue were patent ineligible for the additional reason that they were directed to a method of doing business, because such a method "does not qualify as a 'process' under § 101." *Id.*

[125] Mayo Collaborative Servs. v. Prometheus Labs., 132 S. Ct. 1289, 1294 (2012).

[126] Bilski v. Kappos, 130 S. Ct. 3218, 3230 (2010).

[127] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2358 (majority opinion).

of intermediated settlement on a generic computer."[128] The holding effectively killed all pure business method patents (i.e., all patents that merely recite an abstract economic practice and only add "apply it with a computer"); however, its impact on the larger field of software patents remains to be seen.

Before reaching the particular claims at issue, the Court first announced that the patent subject matter exceptions are motivated by a concern over pre-emption.[129] All exceptions, "laws of nature, natural phenomena, and abstract ideas," are to be treated equally.[130] Additionally, the analysis does not change based on the form of the claim; instead, in *Alice*, the Court analyzed the system claims in the exact same manner as the method claims.[131] As for what types of limitations are "enough" to confer patent eligibility, the Court framed the inquiry in the following manner:

> [I]n applying the § 101 exception, we must distinguish between patents that claim the building blocks of human ingenuity and those that integrate the building blocks into something more, thereby transforming them into a patent-eligible invention. The former would risk disproportionately tying up the use of the underlying ideas, and are therefore ineligible for patent protection. The latter pose no comparable risk of pre-emption,

[128] *Id.* at 2359.

[129] *Id.* at 2355.

[130] *See id.* (indicating that the *Mayo* framework further elaborated in *Alice* applies to all three common law exceptions to patent eligibility).

[131] *See id.* at 2360 (finding the system and computer-readable medium claims to be "no different from the method claims in substance" and thus invalid "for substantially the same reasons"). The Supreme Court's similar treatment of all of the exceptions to patent-eligible subject matter and proclamation that the form of a claim is irrelevant triggered a change in the PTO's examination practices. Following *Mayo*, the PTO issued instructions on how Examiners are to treat process (i.e. method) claims that involve laws of nature/natural correlations. Memorandum from Andrew H. Hirshfeld, Deputy Comm'r for Patent Examination Policy, to Patent Examining Corps 4–9 (July 3, 2012), *available at* http://www.uspto.gov/patents/law/exam/2012_interim_guidance.pdf. These instructions also indicated that product (e.g., system and storage medium) claims are to be analyzed under the *Interim Examination Instructions for Evaluating Subject Matter Eligibility Under 35 U.S.C. § 101* and that process claims directed to an abstract idea should be analyzed under the *2010 Interim* Bilski *Guidance*. *Id.* at 3. Following *Alice*, the PTO issued a new memorandum, which applies to all claims, regardless of whether they are process or product claims, and regardless of whether they incorporate an abstract idea or a law of nature/natural correlation. Memorandum from Andrew H. Hirshfeld, Deputy Comm'r for Patent Examination Policy, to Patent Examining Corps 2 (June 25, 2014), *available at* http://www.uspto.gov/patents/announce/alice_pec_25jun2014.pdf.

> and therefore remain eligible for the monopoly granted under our patent laws.[132]

According to the Court's two-step framework, a court must first "determine whether the claims at issue are directed to one of those patent-ineligible concepts."[133] Next, the court must inquire whether the elements of a claim, either individually or as an ordered combination, have added "enough" to transform the patent-ineligible concept into a patent-eligible application.[134] The Court indicated that its previous references to "inventive concept" and "enough" are synonymous, each referring to "an element or combination of elements that is sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the ineligible concept itself."[135]

Turning to the particular claims at issue, the Court first found that each claim is directed to the abstract idea of intermediate settlement.[136] Specifically, the Court considered the claims in *Alice* to be analogous to the claims in *Bilski*—each constituted "a fundamental economic practice long prevalent in our system of commerce."[137] The Court also described the concept as "a building block of the modern economy" and as "a method of organizing human activity."[138] After determining that the

[132] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2354–55 (citations and internal quotation marks omitted).

[133] *Id.* at 2355.

[134] *Id.*

[135] *Id.* (internal quotation marks omitted).

[136] *Id.* at 2356.

[137] *Id.* In fact, the Supreme Court cited a book from 1896 to underscore this point. *Id.* At least one commentator took notice that the Supreme Court appears to be relying "upon the prevalence of a practice in the prior art" when determining that it is an abstract idea. Rebecca S. Eisenberg, *Symposium: Business Methods as "Abstract Ideas"—Explaining the Opacity of* Alice *and* Bilski, SCOTUSBLOG (June 23, 2014, 1:08 PM), http://www.scotusblog.com/2014/06/symposium-business-methods-as-abstract-ideas-explaining-the-opacity-of-alice-and-bilski. She went on to argue that this indicates that the Supreme Court is modifying § 101 jurisprudence from "preventing the patenting of basic tools because they are too important to leave to the exclusive control of a patent holder" into "prevent[ing] the patenting of modest variations on longstanding practices that are unworthy of patent protection because they add too little to what is already conventional." *Id.* However, the Federal Circuit in *Ultramercial* subsequently held that even an idea that incorporates novel steps may, standing alone, nonetheless be abstract. Ultramercial, Inc. v. Hulu LLC, 772 F.3d 709, 715 (Fed. Cir. 2014) ("We do not agree with Ultramercial that the addition of merely novel or non-routine components to the claimed idea necessarily turns an abstraction into something concrete. In any event, any novelty in implementation of the idea is a factor to be considered only in the second step of the *Alice* analysis.").

[138] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2356.

particular type of intermediate settlement at issue was an abstract idea, the Court refused to further "delimit the precise contours of the 'abstract idea' category."[139] The Court also rejected petitioners' argument that an idea must exist independent of human action in order to qualify for the abstract idea exception.[140]

Next, in step two of its analysis, the Supreme Court asked whether the *Alice* claims add "enough" to the abstract idea, i.e., whether they contain an "inventive concept" to transform this abstract idea into a patent-eligible application.[141] To guide this analysis, the Court reviewed its precedent.[142] In doing so, the Court noted that "well-understood, routine, conventional activities previously known to the industry" were not sufficient to confer patent eligibility.[143] Next, the Court determined that the ubiquity of computers means that "wholly generic computer implementation" and "a generic computer . . . perform[ing] generic computer functions" are insufficient to protect against drafting efforts to monopolize an abstract idea and thus not "enough" to confer patent eligibility.[144] This analysis indicates that the Court sided with Judge Lourie's perspective of computer hardware, the purpose behind the exceptions to § 101, and when computer-implementation may confer patent eligibility.[145] Furthermore, the Court considered the tangible nature of computers and that they "necessarily exist in the physical, rather than purely conceptual realm"

[139] *Id.* at 2357.

[140] *Id.* at 2356–58.

[141] *Id.* at 2357.

[142] *Id.* at 2357–58.

[143] *Id.* at 2359 (internal quotation marks omitted).

[144] *Id.* at 2358–59.

[145] *Compare id.* at 2358 ("[W]holly generic computer implementation is not generally the sort of 'additional feature' that provides any 'practical assurance that the process is more than a drafting effort designed to monopolize the abstract idea itself.'") *with* CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1286 (Fed. Cir. 2013) (Lourie, J., concurring) ("Furthermore, simply appending generic computer functionality to lend speed or efficiency to the performance of an otherwise abstract concept does not meaningfully limit claim scope for purposes of patent eligibility."), *aff'd*, 134 S. Ct. 2347 (2014). In fact, the Supreme Court explicitly cited Judge Lourie's opinion in *CLS Bank* for support for its perspective both on computers and the method claims at issue. *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2358–59. Moreover, this conclusion is consistent with previous Supreme Court precedent, where the Court also viewed the exceptions to patent-eligible subject matter as a substantive limitation meant to protect against overly broad claims. *See* Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289, 1301 (2012) (finding the claims at issue in *Mayo* and the claim previously at issue in *Benson* to be "overly broad" and thus patent-ineligible); Lemley, *Life After* Bilski, *supra* note 101, at 1332–36 (arguing that the substantive limitation theory explains the outcome in each of the major Supreme Court cases dealing with patent eligibility).

irrelevant to its § 101 inquiry, which is in direct conflict with Chief Judge Rader's coarse filter theory.[146]

Finally, the Supreme Court concluded that the system and storage medium claims "are no different from the method claims in substance," and thus are patent-ineligible for substantially the same reasons as the method claims.[147]

As apparent in the aftermath of *Alice*, the Supreme Court's holding has killed pure business method patents, since reciting a fundamental economic practice and instructing a practitioner to implement it with a computer is not patent-eligible.[148] At the same time, *Alice* leaves room for interpretation with regards to other types of software patents, and it remains to be seen how courts and the PTO actual implement the Court's guidance. Most importantly, it remains to be seen whether the Court's endorsement of Judge Lourie's perspective "is the death of hundreds of thousands of patents, including all business method, financial system, and software patents as well as many computer implemented and telecommunication patents."[149]

## III. SOFTWARE AND SOFTWARE PATENTS

With the legal background in place, Part III explores the stakes in the patent eligibility controversy—software innovation. Section III.A explains what software is and how it is designed, while Section III.B provides an overview of the debate regarding whether software patents promote or discourage innovation.

[146] *Compare Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2358–59 ("The fact that a computer 'necessarily exists in the physical, rather than purely conceptual, realm,' . . . is beside the point. There is no dispute that a computer is a tangible system (in § 101 terms, a 'machine')," but, "if that were the end of the § 101 inquiry," "[s]uch a result would . . . eviscerate[e] [the common law exceptions to patent eligibility.]") *with CLS Bank Int'l*, 717 F.3d at 1305 (Rader, C.J., concurring in part and dissenting in part) ("If tying a method to a machine can be an important indication of patent eligibility, it would seem that a claim embodying the *machine itself*, with all its structural and functional limitations, would rarely, if ever, be an abstract idea."); *id.* at 1314 (Moore, J., dissenting in part) ("[N]o contortion of the term 'abstract idea' can morph this physical system into an abstract idea.").

[147] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2360.

[148] *See id.* at 2358–59.

[149] *CLS Bank Int'l*, 717 F.3d at 1313 (Moore, J., dissenting in part).

## A. DEFINING SOFTWARE

Modern software has become vastly more complex than the simple flowcharts at issue in *Benson*.[150] Today, it is impractical for any practitioner to sit down and write all the individual software instructions that make up even a simple program.[151] As such, software has evolved from a flowchart model, where numerical operations are serially performed on a particular set of inputs, into a layered model, where different components of software blindly rely on each other to provide certain functionalities.[152] Ironically, given the subject matter of this Note, this layered approach to software design is called "abstraction."[153] Abstraction allows for easier design and debugging of software, since practitioners can design, implement, debug, and subsequently upgrade an individual layer without having to simultaneously modify other layers.[154] Similarly, practitioners can import an improved version of a given layer without having to modify existing layers.[155]

To facilitate discussion and avoid confusion, the remainder of this Note will discuss software with the following exemplary set of software layers:

[150] *See* PATTERSON, *supra* note 1, at 10 ("A typical application, such as a word processor or a large database system, may consist of millions of lines of code and rely on sophisticated software libraries that implement complex functions in support of the application.").

[151] *See id.* at 10, 21 ("This principle . . . is the way both hardware designers and software designers cope with the complexity of computer systems.").

[152] *See id.* at 20 ("To go from a complex application to the simple instructions involves several layers of software that interpret or translate high-level operations into simple computer instructions."), 20–21 ("Typically, the operating system will encapsulate the details of doing I/O, allocating memory, and other low-level system functions so that application programmers do not need to worry about such details."); *see also Android Interfaces*, ANDROID DEVELOPERS, http://source.android.com/devices/index.html (last visited Jan. 30, 2015) (Application developers only need to concern themselves with one out of five software layers, as all other communication between these layers "is hidden from the developer and things appear to 'just work.'").

[153] *See* PATTERSON, *supra* note 1, at 20 ("The use of such layers, or *abstractions*, is a principal technique for designing very sophisticated computer systems."); *see also Android Interfaces*, *supra* note 152 (discussing a hardware *abstraction* layer (HAL)).

[154] *See* Oracle Corp., *What Is an Object?*, THE JAVA™ TUTORIALS, http://docs.oracle.com/javase/tutorial/java/concepts/object.html (last visited Feb. 10, 2015) (discussing some of the benefits of "object oriented programming," which is a type of layering used in software design).

[155] *See id.*

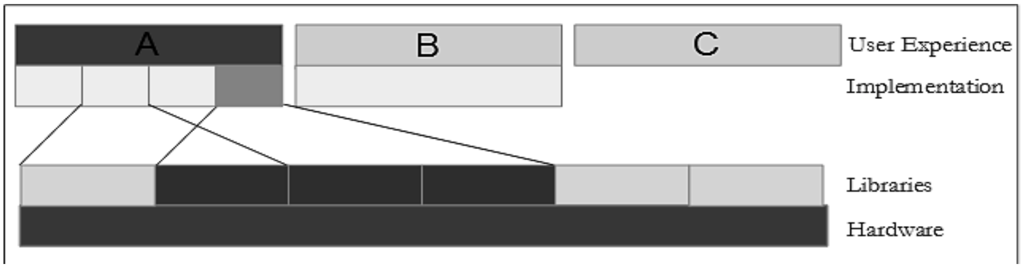

**Figure 1: User experience A might be a video game, user experience B might be a ranking algorithm, and user experience C might be the user interface of a word processing program.**

To give a concrete example before diving into the details of each layer, assume that these layers correspond to software and hardware found within a cellular phone, and that the highlighted blocks are all of the software and hardware components involved in allowing a user to play a video game (i.e., the components involved in providing user experience *A*). A practitioner designing software for this cellular phone, such as an application developer, would set out to create the video game (i.e., user experience *A*). To accomplish this, the application developer would only have to write the code corresponding to a single implementation.[156] In writing this code, the application developer would be able to call functions provided by one or more components of the library layer (e.g., the Android operating system) for common tasks such as storing data, receiving user input, triggering interruptions, or displaying graphics to the user.[157] The application developer does not need to know how the library layer accomplishes each of these tasks. Instead, she only needs to know what inputs the library layer requires and what it promises as its output or result (i.e., the application programming interface).[158] The library layer, in turn, is the only layer that then actually interacts with the hardware.[159] This allows the same video game to function on any cell phone or tablet, regardless of its hardware, that has a library layer with the same

[156] *See Android Interfaces*, *supra* note 152 (an application developers only needs to write the application framework layer). The application developer may, however, decide to break up the implementation layer into multiple layers to further simplify development using, e.g., object oriented programming. *See* Oracle Corp., *supra* note 154.

[157] *See Android Interfaces*, *supra* note 152 ("The Binder Inter-Process Communication mechanism allows the application framework to cross process boundaries and call into the Android system services code.").

[158] *See id.*

[159] *See id.* ("The [hardware abstraction layer (HAL)] serves as a standard interface that allows the Android system to call into the device driver layer while being agnostic about the lower-level implementations of your drivers and hardware.").

application programming interface (e.g., another cell phone that runs the Android operating system).[160] Thus, the implementation layer is hardware agnostic, which allows for hardware independence (i.e., the same piece of software can be executed on other hardware platforms).[161] Hardware independence is a desirable goal in software development, as it allows the application developer to reach a wider consumer base than if the program could run only on specific devices.[162]

The user experience is what the user actually expects a piece of software to accomplish—it is the end-result of running the piece of software. For example, a user experience might be the interface displayed in a word processing program and the set of features available to the user, or the user experience might involve storing files selected by a user to the cloud. The patents at issue in *Alice* illustrate another example of a user experience—blocking transactions that a particular user cannot afford during the time period after a contract is formed but before it matures, and then settling these transactions at the end of the day.[163]

If the user experience layer is *what* a piece of software accomplishes, then the next layer, the implementation layer, is the first layer that describes *how* that piece of software accomplishes this particular result. Now, a given user experience might have multiple different implementations that lead to the same result, as is the case with user experience *A*. Alternatively, only a single implementation might be possible or currently known, as is the case with user experience *B*. Finally, while in theory every user experience has an underlying implementation, in some cases the implementation might be straightforward and there might be a direct mapping between the user experience and the libraries utilized by the implementation. Described using legal terminology, such an implementation would be an inherent part of the user experience or at most involve "well-understood, routine, conventional activities." To highlight this situation, user experience *C* is drawn with no

[160] *See id.*; *see also Device Compatibility*, ANDROID DEVELOPERS, http://developer.android.com/guide/practices/compatibility.html (last visited Jan. 30, 2015) ("Android is designed to run on many different types of devices, from phones to tablets and televisions.").

[161] *See Android Interfaces*, *supra* note 152; *see also Device Compatibility*, *supra* note 160.

[162] *See Device Compatibility*, *supra* note 160; *see also* PATTERSON, *supra* note 1, at 20–21 ("[The] abstract interface [between the hardware and low-level software] allows many *implementations* of varying cost and performance to run identical software.").

[163] *See* CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1285 (Fed. Cir. 2013) (en banc) (Lourie, J. concurring), *aff'd*, 134 S. Ct. 2347 (2014).

implementation at all. The claims at issue in *Alice* fall into this third category—any implementation necessary to achieve the user experience claimed in the *Alice* patents is already directly reflected in the claimed user experience, and thus inherently part of the user experience or at most constitutes "well-understood, routine, conventional activities."[164]

The next layer, the library layer, consists of software that might be useful to a number of different implementations and that thus is maintained as part of a central library accessible by all of these different implementations. Grouping such universally useful software into a single set of libraries saves storage space by eliminating redundant code, allows the implementation layers to be hardware agnostic, and facilitates improvements by allowing practitioners to update a single library instead of individually modifying each implementation that uses this library.[165] In the context of personal computer ("PC") architecture, the library layer may include the operating system that bridges the gap between applications and the hardware.[166] Other functionalities that are frequently included in the library layer are media players, compression algorithms, display formats, user interface elements, and network connection managers.[167]

Finally, there is the hardware itself. The line between software and hardware is a fluid one, however, since anything coded in software can also be implemented directly in hardware.[168] The decision of whether to implement a particular feature in hardware or software is merely a design choice—it is easier to implement features in software, but hardware is capable of performing the same processing faster.[169]

[164] *See* Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2359 (2014).

[165] *See* Oracle Corp., *supra* note 154.

[166] *See* PATTERSON, *supra* note 1, at 10.

[167] *See Class Index*, ANDROID DEVELOPERS, http://developer.android.com/reference/classes.html (last updated Feb. 12, 2015, 7:42 PM) (listing classes within the Android operating system that are available to developers for use).

[168] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1306 (Fed. Cir. 2013) (en banc) (Rader, C.J., concurring in part and dissenting in part); *see also* Paul Graham, *Are Software Patents Evil?*, PAULGRAHAM (Mar. 2006), http://www.paulgraham.com/softwarepatents.html ("Since software patents are no different from hardware patents, people who say 'software patents are evil' are saying simply 'patents are evil.'").

[169] *See Hardware or Software Video Decoder?*, ANDROIDCENTRAL, http://forums.androidcentral.com/google-nexus-7-tablet-2012/240201-hardware-software-video-decoder.html (last updated Jan. 18, 2014, 12:06 PM) (online forum where practitioners discuss the benefits and drawbacks of using a hardware versus software video decoder). The line between software and hardware is further blurred because many features that, from the perspective of the main processor, are performed by hardware are actually still implemented as software running on a separate and specialized digital signal processor. *See A Beginner's Guide to Digital Signal Processing*, ANALOG

## B. THE ROLE OF SOFTWARE PATENTS

The trade-off in determining the appropriate level of patent protection is easily stated—the benefit of encouraging innovation through the grant of a monopoly versus the cost of that monopoly to consumers and follow-on innovators—but difficult to measure.[170] This balance becomes especially delicate in a cumulative field, such as software, where components need to interact and build on each other to accomplish a particular user experience.[171] In cumulative fields, allowing overly broad patents is likely to harm innovation in the future more than it encourages innovation in the present[172]—the very concern the Supreme Court is attempting to solve via its § 101 jurisprudence.[173]

Moreover, software is especially susceptible to overly broad patents. Each software layer individually is fully functional—it can be described and claimed by stating only its desired outcome.[174] This functional aspect of software claiming has led to overly broad software patents. The existence of such broad functional claims is especially troubling if a practitioner who creates a particular user experience is able to claim it by merely reciting the user experience and appending "apply it with a computer" (i.e., a claim reciting only the elements highlighted below). The scope of such a claim is illustrated below:

DEVICES, http://www.analog.com/en/content/beginners_guide_to_dsp/fca.html (last visited Feb. 10, 2015).

[170] *See* Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329, 1340 (1987).

[171] *See* Robert P. Merges & Richard R. Nelson, *On the Complex Economics of Patent Scope*, 90 COLUM. L. REV. 839, 908–911 (1990).

[172] *See id.*

[173] *See* Lemley, *Life After* Bilski, *supra* note 101, at 1317–18; *supra* Section II.B.

[174] Kevin E. Collins, *Patent Law's Functionality Malfunction and the Problem of Overbroad, Functional Software Patents*, 90 WASH. U. L. REV. 1399, 1402–03 (2013).

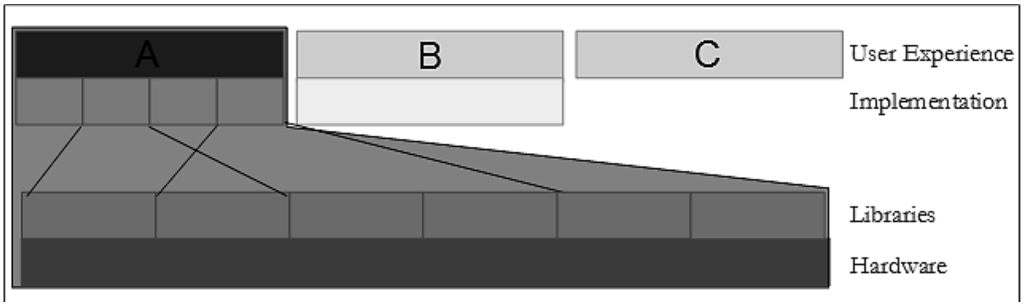

**Figure 2: The scope of a claim reciting only a user experience implemented with a computer is represented by the transparent overlay.**

While the inventor may be the first to provide user experience *A*, what she really built is only a single implementation that leads to user experience *A*. However, by framing a claim as directed to user experience *A* "applied with a computer," she is able to lay claim to and prevent others from competing with her in providing user experience *A*, regardless of how little a potential competitor's implementation has in common with her own implementation.[175] The scope of her claim is thus not commensurate with her "practical, real-world contribution," and her patent is therefore overly broad.[176]

Software patents also frequently present notice externalities—additional costs that arise from unclear and not easily discoverable property boundaries.[177] In general, intangible property rights, such as patents, are more prone to notice externalities than real property rights, since intangible properties can overlap in scope and there are no physical boundaries.[178] Moreover, software patents are especially susceptible to notice concerns due to uncertainty regarding their claim scope[179] caused by the amorphous claim language often found in software patents[180] and the lack of a common nomenclature.[181] This problem is further compounded by the difficulty of searching for potentially infringed patents and the

[175] Mark A. Lemley, *Software Patents and the Return of Functional Claiming*, 2013 WIS. L. REV. 905, 907–08 (2013) (By obtaining broad functional claims, patentees "effectively capture[] ownership not of what they built, but of anything that achieves the same goal, no matter how different it is.").

[176] Lemley, *Life After* Bilski, *supra* note 101, at 1317.

[177] Peter S. Menell & Michael J. Meurer, *Notice Failure and Notice Externalities*, 5 J. LEGAL ANALYSIS 1, 1–10 (2013).

[178] *Id.* at 2, 15–17.

[179] *Id.* at 33.

[180] *Id.* at 20.

[181] *Id.* at 36.

dubious validity of many software patents.[182] In fact, it is reasonable to assume that every new software startup infringes patents without knowing.[183] Lack of notice, in turn, creates additional costs to other practitioners in the industry, such as costs of searching for and analyzing potentially infringed patents, and litigation costs if licensing fails,[184] thereby discouraging innovation.[185]

This notice problem is even more disconcerting in the case of overly broad software patents, such as a patent that claims a library divorced from any particular user experience or implementation:

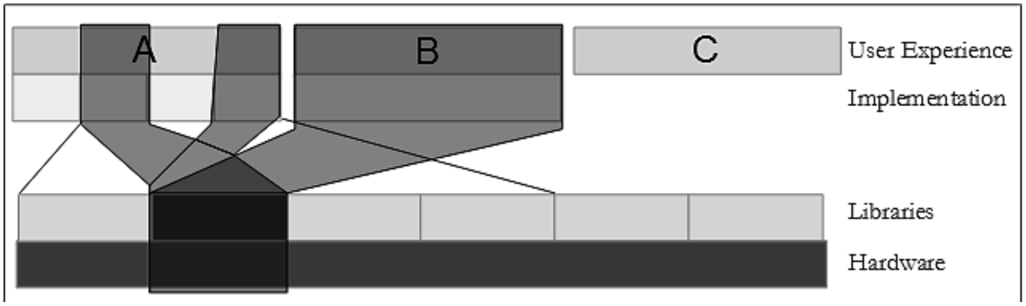

**Figure 3: The scope of a claim reciting only a library implemented with a computer is represented by the transparent overlay.**

While a practitioner may have initially implemented the claimed library as part of software that achieves one user experience (e.g., user experience *A*) and described the library in this manner within the patent's specification, the claim illustrated above would also be infringed by other practitioners working in unrelated fields (e.g., user experience *B*). Thus, another practitioner wanting to perform a freedom to operate search (i.e., a search performed before entering a field to determine whether there are any incumbent IP rights) before creating a software product would not just have to contend with the inherent vagueness of software claims, but additionally would be unable to confine her search to any particular area of technology (i.e., user experience). For example, assume there existed a patent on the Fast Fourier Transform.[186] Further, assume that in its

[182] *Id.* at 5–6.

[183] Graham, *supra* note 168 ("Don't waste your time worrying about patent infringement. You're probably violating a patent every time you tie your shoelaces.").

[184] Menell, *Notice Failure*, *supra* note 177, at 9–10.

[185] *Id.* at 39.

[186] The Fast Fourier Transform is an algorithm for converting a signal (e.g., a sound wave) back and forth between the time domain (e.g., the vibrations in the air a human ear picks up as sound) and the frequency domain (e.g., what combination of frequencies or notes are present in the sound). *See* Phil Burk et al., *Chapter 3: The*

specification this patent presents the algorithm in the context of sound processing, but then claims the algorithm independent of any sound processing (i.e., the patent only claims the library layer without any accompanying user experience). Even if a practitioner working on a new magnetic resonance imagining ("MRI") machine were to invest an impractically large amount of time and money to perform a complete freedom to operate search within the field of medical imaging, she would still never discover the Fast Fourier Transform patent, since the only concrete use mentioned in the patent is sound processing.

Despite these negative aspects, commentators have argued that at least some software innovation is appropriately encouraged using patent protection. Developing a new software algorithm is not inherently different from other research and development investments, and software innovation should thus be encouraged via patent protection for the same reasons as innovation in these other fields.[187] Additionally, software, when executed by a computer, is a physical process performed by physical circuits and thus not different from or any more abstract than the more traditional inventions that have historically enjoyed patent protection.[188] At the same time, other commentators have argued that the software industry is fundamentally different from other industries, since software consumers are also likely to build upon an invention, leading to follow-on software innovation.[189] Introducing patent protection into such an ecosystem not only raises the price of the original invention to consumers, but also decreases the number of consumers improving the original invention, and thus decreases follow-on innovation.[190]

In the end, resolving the debate over whether or not software innovation is appropriately encouraged using patent protection is outside the scope of this Note. In fact, it is difficult if not impossible to determine whether any particular software innovation deserved patent protection,[191]

*Frequency Domain*, MUSIC AND COMPUTERS: A THEORETICAL & HIST. APPROACH, http://music.columbia.edu/
cmc/musicandcomputers/chapter3/03_04.php (last visited Feb. 8, 2015). The Fast Fourier Transform has allowed advances in a variety of fields, including medical imaging, wireless communication, sound processing, and many more. *See, e.g.*, Newell, *supra* note 17, at 1028.

[187] Chisum, *supra* note 17, at 1015; *see also* Merges, *Go Ask* Alice, *supra* note 30 (arguing that the Google page rank patent "ought to survive Section 101 analysis").

[188] *See See* PATTERSON, *supra* note 1, at 329 fig.4.24.

[189] Newell, *supra* note 17, at 1033–34.

[190] *See id.*

[191] Before concluding that any particular innovation was appropriately encouraged using patent protection, the benefit to society of a patent on this innovation must be

and this Note does not attempt to make any such determination. However, software innovation is not categorically excluded from patent protection under current Supreme Court precedent.[192] Thus, if *some* software innovation remains patent-eligible, it becomes necessary to distinguish between software innovation that is *less* deserving of patent protection and software innovation that is *more* deserving of patent protection. Given that the patent-eligible subject matter exceptions serve as a substantive limitation to protect against overly broad patents,[193] this entails distinguishing between software patents based on their claim scope, instead of finding both claims that are commensurate in scope with the inventor's contribution and overly broad claims on business methods equally patent-ineligible.

## IV. INTERPRETING *ALICE*

While the central holding in *Alice* is relevant, it came as no surprise—instructing a practitioner to apply an abstract idea with a computer is not "enough" to confer patent eligibility.[194] The full effect of *Alice*, however, depends on what lower courts and the PTO do with the Court's guidance. Part IV provides an overview of the different interpretations of *Alice* used by courts, the PTO, and commentators when performing each of the two steps of the *Alice* framework, and proposes an alternative interpretation for each step. Section IV.A discusses step one of this analysis, and Section IV.B discusses step two.

### A. STEP ONE—WHAT IS AN ABSTRACT IDEA?

Courts interpreting step one of the *Alice* framework differently will identify different abstract ideas even in the same claim. Section IV.A discusses three different tests for step one that have emerged since the *Alice* decision—a "mathematical formula" test employed by some courts and the PTO (Section IV.A.1), a pen-and-paper test employed by other

weighed against the social costs of the granted monopoly to determine whether society truly benefited from the patent's existence. Menell, *Tailoring Legal Protection for Computer Software*, *supra* note 170, at 1340. However, determining if any particular innovation would have occurred but-for patent protection and quantifying the social benefits and costs of a patent on that particular innovation is difficult if not impossible.

[192] Bilski v. Kappos, 130 S. Ct. 3218, 3226–27 (2010); *supra* Section I.C; *see also* Ca. Inst. of Tech. v. Hughes Commc'ns Inc., No. 2:13-cv-07245-MRP-JEM, 2014 WL 5661290, at *7–8 (C.D. Cal. Nov. 3, 2014) (arguing that Congress "contemplate[d] the existence of software patent" when it passed the America Invents Act and that "the Supreme Court . . . implicitly endorsed the patentability of software").

[193] *See supra* Section II.B.

[194] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2358 (2014).

courts (Section IV.A.2), and a "multiple abstract ideas" test proposed in this Note (Section IV.A.3).

### 1. *The Mathematical Formula Test*

Following *Alice*, the Federal Circuit in *DDR Holdings, LLC v. Hotels.com, L.P.*,[195] and the PTO[196] apparently have tried to limit the impact of *Benson* on software by applying what this Note refers to as a "mathematical formula test" for identifying a patent-ineligible abstract idea—within the context of algorithms, only "mathematical" algorithms, relationships, and formulas[197] constitute abstract ideas. Thus, in *DDR Holdings, LLC*, the claims at issue recited features of an algorithm, including "a computer server" that receives "a signal" indicating that a link was selected, "automatically identif[ies]" the source page corresponding to the selected link, "automatically retrieve[s] . . . stored data corresponding to the source page," and, "using the data retrieved, automatically generate[s] and transmit[s] . . . a second web page."[198] However, the court noted that "identifying the precise nature of the abstract idea is not as straightforward as in *Alice*," in part because the "asserted claims do not recite a *mathematical* algorithm."[199] Similarly, the PTO issued instructions to examiners on analyzing patent eligibility under § 101 in view of *Alice*.[200] These instructions set the standards examiners apply when evaluating a patent application and thus will affect what software patents will be issued in the future.[201] According to these instructions, "[f]undamental economic practices," "[c]ertain methods of organizing human activities," "[a]n idea of itself" and "*[m]athematical* relationships/formulas" are abstract ideas.[202] Thus, by avoiding use of the more expansive term of "algorithm" without any qualifier, both the *DDR Holdings* court and the PTO appear to be drawing a distinction between a "general algorithm" and a "mathematical algorithm" or "[m]athematical relationships/formulas" and implying that only claims reciting the latter but not only the former are directed to a patent-ineligible abstract idea.[203]

[195] 773 F.3d 1245, 1257 (Fed. Cir. 2014).

[196] Hirshfeld, Memorandum dated June 25, 2014, *supra* note 131, at 2–3.

[197] Lacking any indication to the contrary, and for the sake of simplicity, this Note assumes that mathematical algorithms, relationships, and formulas are synonymous and can all be described as "mathematical formulas."

[198] *DDR Holdings, LLC*, 773 F.3d at 1249–50.

[199] *Id.* at 1257 (emphasis added).

[200] Hirshfeld, Memorandum dated June 25, 2014, *supra* note 131, at 2.

[201] *Id.* at 1–3.

[202] *Id.* at 2–3 (emphasis added).

[203] *See id.*; *DDR Holdings, LLC*, 773 F.3d at 1249.

This distinction between patent-eligible "general algorithms" and patent-ineligible "mathematical formulas" traces back to *Application of Freeman*, where the C.C.P.A. distinguished between a "mathematical algorithm" of the type held patent-ineligible in *Benson* and "the term 'algorithm' as a term of art in its broad sense."[204] According to *Freeman*, some types of data processing software, such as "processing [a] hierarchical tree structure and spatially relating the various characters to be displayed," might be an "algorithm," but it is not a "mathematical algorithm" of the type deemed patent-ineligible in *Benson*.[205] Thus, under *Freeman*, while a mathematical algorithm is an abstract idea, the same need not be the case for all algorithms. This distinction is significant to software, where every process consists of a series of steps for manipulating data and thus necessarily is an "algorithm," but where arguably many types of data manipulation do not constitute "mathematical algorithms."[206]

However, the mathematical formula test is not consistent with *Alice*'s characterization of the abstract idea at issue in *Benson*. In *Mayo*, the Court characterized *Benson* as holding that "a *mathematical process* for converting binary-coded decimal numerals into pure binary numbers," "a *mathematical* principle," and a "*mathematical* formula" are not patent-eligible[207] However, in *Alice*, the Court described the abstract idea at issue in *Benson* as "an *algorithm* for converting binary-coded decimal numerals into pure binary form."[208] Assuming that the Court selected the language in both *Mayo* and *Alice* purposefully, and was aware of the distinction lower courts made between an algorithm and a mathematical formula, this change

[204] 573 F.2d 1237, 1245–46 (C.C.P.A. 1978). While subsequent decisions overruled the *Freeman-Walter-Abele* test, which was based in part on this case, these subsequent decisions did not address the discussion of general algorithms versus mathematical algorithms in *Freeman*. *See* State St. Bank & Trust Co. v. Signature Financial Grp., Inc., 149 F.3d 1368, 1373–74 (Fed. Cir. 1998).

[205] Application of Freeman, 573 F.2d at 1246.

[206] Andrei Iancu & Peter Gratzinger, *Machines and Transformations: The Past, Present, and Future Patent ability of Software*, 8 NW. J. TECH. & INTELL. PROP. 247 at *46 (2010) ("By narrowing the forbidden computational algorithms to 'mathematical' algorithms, *Freeman* appeared to narrow dramatically the type of information-intensive processes, such as software patents, that would be excluded under the Supreme Court's holding in *Benson*."); *see also* Merges, *Go Ask* Alice, *supra* note 30 ("By saying in effect that algorithms are a species of abstract idea, the Court invited all mischief. The entire shelf full of discredited cases on the metaphysics of what is and is not an algorithm must now be dusted off.").

[207] Mayo Collaborative Servs. v. Prometheus Labs., 132 S. Ct. 1289, 1301 (2012) (emphasis added).

[208] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014) (emphasis added).

indicates that the Court now considers all algorithms, whether mathematical or not, patent-ineligible abstract ideas.

Additionally, using the mathematical formula test presents practical problems. Deciding whether a claim directed to software covered a patent-eligible "algorithm" or a patent-ineligible "mathematical algorithm" might be difficult if not impossible.[209] A computer, by its very nature, can only do one of a few things—read a number from memory or an input, write a number to memory or an output, jump to another instruction based on a comparison of two numbers, or perform an arithmetic or logical operation on two numbers.[210] Of these, the operations that actually modify data and thus form the heart of software—arithmetic or logical operations—fall squarely in the category of "mathematical formulas."

### 2. *The Pen-and-Paper Test*

Courts, even after *Alice*, have continued to apply the "pen-and-paper" test,[211] which the Federal Circuit formulated in its pre–*Alice CyberSource*

[209] *In re* Warmerdam, 33 F.3d 1354, 1359 (Fed. Cir. 1994) ("The difficulty is that there is no clear agreement as to what is a 'mathematical algorithm,' which makes rather dicey the determination of whether the claim as a whole is no more than that."); Newell, *supra* note 17, at 1025 ("[A]ny attempt to find a helpful or cutting distinction between mathematics and nonmathematics, as between numerical or nonnumerical, is doomed."); Lee, *Software Is Just Math*, *supra* note 1 (arguing that "all software that has ever been written" consists only of "mathematical algorithms" (emphasis removed)).

[210] *See* PATTERSON, *supra* note 1, at 78 (listing all instructions an exemplary processor in a computer is capable of executing); *see also* Lee, *Software Is Just Math*, *supra* note 1 (arguing that "all software that has ever been written" consists only of "mathematical algorithms" (emphasis removed)). In fact, all data processing that a computer could possibly perform amounts to no more than a mathematical operation. A computer contains a processor consisting of digital circuits. *See* PATTERSON, *supra* note 1, at 329 fig.4.24; David Patterson & John L. Hennessy, *Chapter 4: The Processor*, COMPUTER ORGANIZATION AND DESIGN: THE HARDWARE/SOFTWARE INTERFACE 30 (2009) http:// booksite.mkp.com/patterson/lec.php (requires logging in). Every time a particular software instruction has to be executed, voltages representing the instruction's input numbers are applied to a first set of wires within the processor, and, after the digital circuit reaches equilibrium, the voltages present on a second set of wires represent the instruction's output numbers. *See Chapter 4: The Processor*, *supra*, at 8–9, 23. As such, a computer is incapable of manipulating data in any manner except by operating on voltages corresponding to numerical representations of the data.

[211] *See, e.g.*, Planet Bingo, LLC v. VKGS LLC, 576 F. App'x 1005, 1007–08 (Fed. Cir. 2014); Cogent Med., Inc. v. Elsevier Inc., Nos. C-13-4479-RMW, C-13-4483, C-13-4486, 2014 WL 4966326, at *4 (N.D. Cal. Sept. 30, 2014); Loyalty Conversion Sys. Corp. v. Am. Airlines, Inc., No. 2:13-CV-655, 2014 WL 4364848, at *10 (E.D. Tex. Sept. 3, 2014); Walker Digital, LLC v. Google, Inc., No. 11-318-LPS, 2014 WL 4365245, at *9 (D. Del. Sept. 3, 2014); Comcast IP Holdings I, LLC v. Sprint Commc'ns Co. L.P., No. 12-205-RGA, 2014 WL 3542055, at *4–5 (D. Del. July 16,

*Corp. v. Retail Decisions, Inc.* decision.[212] In *CyberSource*, the Federal Circuit relied on *Benson* and *Flook* and held that "methods which can be performed mentally, or which are the equivalent of human mental work, are unpatentable abstract ideas—the 'basic tools of scientific and technological work' that are open to all."[213] This includes methods that "can be performed in the human mind, or by a human using a pen and paper." [214]

The pen-and-paper test is appealing due to its ease of application and clear delineation of what constitutes an abstract idea.[215] However, courts still need to be vigilant of false positives—e.g., "a human being could perform the calculations that would yield the value of a parity bit" (i.e., a bit added to the end of a file or transmission that is used to detect whether the file or transmission has been corrupted), but this exercise would not actually produce a parity bit, since the result of the human calculation cannot be used to detect "the corruption of data during transmission."[216] Except for these false positives, the test also has the potential to render all software executed by a general-purpose computer patent-ineligible, since"[e]verything done by a computer can be done by a human."[217]

A fundamental problem with the *CyberSource* inquiry is that it does not take into account how detailed or specific an algorithm is or how many components it has; instead, an algorithm is a single abstract idea if it can

2014); DietGoal Innovations LLC v. Bravo Media LLC, 33 F. Supp. 3d 271, 284 (S.D.N.Y. 2014).

[212] *See* 654 F.3d 1366, 1371 (Fed. Cir. 2011).

[213] *Id.*

[214] *Id.* at 1372.

[215] *See Walker Digital, LLC*, 2014 WL 4365245, at *9 (doing a direct comparison of claim elements to human activity in order to identify the abstract idea at issue).

[216] Ca. Inst. of Tech. v. Hughes Commc'ns Inc., No. 2:13-cv-07245-MRP-JEM, 2014 WL 5661290, at *16 (C.D. Cal. Nov. 3, 2014). The court in *California Institute of Technology* argued that the pen-and-paper test is instead "a stand-in" for determining whether "humans engaged in the same activity long before the invention of computers." *Id.* However, since the Federal Circuit analyzed the mathematical expressions at stake in *Benson* and *Flook* when it formulated the pen-and-paper test, the test is more appropriate described as "a stand-in" for determining whether a process constitutes a patent-ineligible "algorithm." *See CyberSource Corp.*, 654 F.3d at 1371–72.

[217] CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1306 n.7 (Fed. Cir. 2013) (en banc) (Rader, C.J., concurring in part and dissenting in part), *aff'd*, 134 S. Ct. 2347 (2014). Now, there are pieces of hardware in a computer other than the processor that are capable of actions that a human cannot do, but these are not a necessary part of a general-purpose computer. *See Device Compatibility*, *supra* note 160 (discussing availability of pressure sensors on some, but not all, devices). Instead, it is desirable for software to not rely on such specialized hardware and thus remain hardware agnostic. *See supra* Section III.A.

be performed in the human mind or by a human using pen and paper.[218] This categorical approach is at odds with the Supreme Court's view that § 101 is a substantive limitation designed to protect against overly broad patents.[219] Specifically, identifying an "overly" broad claim is not a simple black-or-white determination, but a nuanced exercise of discretion. Any invention can be characterized at progressively narrower levels of abstraction, and a court needs to balance the benefit of creating incentives for innovation against the risk of preempting subsequent improvements when drawing the line between abstract idea and patentable invention.[220] As an illustration, assume that a court is analyzing a set of progressively narrower claims, all covering the same piece of software that performs a function capable of being performed by a human. Here, the pen-and-paper test does not allow a court to draw a line between overly broad claims and claims that are sufficiently narrow so that they ought to be patent-eligible under the Supreme Court's substantive limitation theory of § 101. Instead, the court is forced to either hold all claims patent-eligible or patent-ineligible—a simple black-or-white determination that is more consistent with Chief Judge Rader's view of § 101 as a coarse filter,[221] a view the Supreme Court did not endorse in *Alice*.[222]

Moreover, lumping all of these algorithms into a single abstract idea can lead to a court defining an absurdly narrow abstract idea and thus losing sight of the overarching policy goal of protecting "the *basic* tools of scientific and technological work."[223] This is not just hyperbole—the court in *McRO, Inc.* defined the abstract idea at stake as "automated rules-based use of morph targets and delta sets for lip-synchronized three-dimensional animation."[224]

### 3. *The Multiple Abstract Ideas Test*

Even fully accepting the extent of the Supreme Court's holding in *Benson* that an algorithm is an abstract idea, there is no reason why a claim

[218] *See Walker Digital, LLC*, 2014 WL 4365245, at *9 (finding the entire claimed process to be a single abstract idea).

[219] *See* Lemley, *Life After* Bilski, *supra* note 101, at 1317; *supra* Section II.B.

[220] *See* Tun-Jen Chiang, *The Levels of Abstraction Problem in Patent Law*, 105 NW. U. L. REV. 1097, 1099–1100 (2011); Collins, *Patent Law's Functionality Malfunction*, *supra* note 174, at 1464–65.

[221] *See supra* Section II.A.

[222] *See supra* Section II.B.

[223] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2354 (2014) (emphasis added).

[224] McRO, Inc. v. Activision Publ'g, Inc., No. CV 14-336-GW(FFMx), 2014 WL 4759953, at *10 (C.D. Cal. Sept. 22, 2014).

directed to an algorithm must contain only a single detailed abstract idea, instead of multiple broader abstract ideas. Instead, this Note proposes that courts should apply a "multiple abstract ideas" test by separating a claimed software method along the layers that make up the software[225] and defining the algorithms making up each layer as separate abstract ideas. Thus, under the multiple abstract ideas test, the same steps of a claim would be deemed patent-ineligible as under the pen-and-paper test, but these steps would be split up into multiple separate abstract ideas. This allows courts to stay true to Supreme Court precedent regarding the general patent-ineligibility of algorithms, instead of attempting to artificially cabin *Benson*'s holding to mathematical formulas, and to use the conceptually appealing pen-and-paper test, while still making a nuanced policy determination on how narrowly any individual abstract idea should be defined. For example, the court in *McRO, Inc.* could have identified two separate abstract ideas—a user experience ("automated lip synchronization using morph targets and delta sets") and an implementation ("rules-based setting of morph targets and delta sets").[226]

## B. STEP TWO—WHAT IS "ENOUGH" TO CONFER PATENT ELIGIBILITY?

Since *Alice*, courts and commentators have also taken divergent approaches to step two of the *Alice* framework. At least one court argued that *Alice* revived *Flook* (discussed in Section IV.B.1). Multiple courts argued that only physical limitations are "enough" to confer patent eligibility (discussed in Section IV.B.2). Finally, this Note and at least one court argue that even a second abstract idea ought to confer patent eligibility onto a first abstract idea (discussed in Section IV.B.2).

### *1.* Alice *Revived* Flook

At least one court has taken the position that *Alice* wholly endorsed and revived *Flook*'s framework. In a series of opinions, the *McRO, Inc.* court described *Alice*'s two-step test for abstraction as analogous to the famous test for obscenity—"I know it when I see it."[227] The court

[225] *See supra* Section III.A.

[226] *See McRO, Inc.*, 2014 WL 4759953, at *10. The author acknowledges that claims encompassing even a *single* abstract idea have already presented enough of a headache to patent practitioners and jurists. However, *infra* Section IV.B.3 will explain how the multiple abstract ideas test can be used to decide whether any individual software innovation is patent-eligible.

[227] *McRO, Inc.*, 2014 WL 4759953, at *5 (quoting *Jacobellis v. Ohio*, 378 U.S. 184, 197 (1964) (Stewart, J., concurring)).

compared each of the asserted claims against the admitted prior art and identified "the use of rules, rather than artists, to set the morph weights and transitions between phonemes" as the sole novel aspect of the claims.[228] The court then concluded that the asserted claims are patent-ineligible because they contain "an abstract idea at the point of novelty."[229]

Additionally, at least one commentator and at least one other court, while not explicitly using *Flook*'s framework, adopted aspects of *Flook*'s analysis by referencing novelty and non-obviousness in their discussion of § 101. Professor Merges argued that "*novel* hardware elements are simply not a prominent feature of many software patents" and that "the second step of the [Alice] framework seems duplicative or at least highly similar to the traditional *novelty* and *non-obviousness tests* of U.S. patent law."[230] Similarly, the Federal Circuit in *Ultramercial, Inc. v. Hulu LLC* found the claims at issue patent-ineligible because they were "not tied to any particular *novel* machine or apparatus, only a general-purpose computer."[231] While neither Professor Merges nor the *Ultramercial* court explicitly stated that an inventive concept must be a novel, non-abstract additional limitation, as would be necessary if *Alice* had revived *Flook*'s framework,[232] both the article and the decision indicate that practitioners and jurists still at least partially resort to *Flook*'s point-of-novelty framework when analyzing patent eligibility under *Alice*.

*McRO, Inc.*'s revival of *Flook* by requiring novelty before a limitation can be "enough" to confer patent eligibility is inconsistent with the shift in language used by the Supreme Court in *Alice*.[233] Specifically, *Alice*, in contrast to *Flook* and *Mayo*, avoided conflating "well-understood, routine, conventional activities" insufficient for patent eligibility with activities that are not novel or obvious. As recently as *Mayo*, the Court stated that "[p]urely 'conventional or *obvious*' '[pre]-solution activity'" is insufficient to confer patent eligibility.[234] In contrast, the Court in *Alice* never used the

[228] *Id.* at *10–11.

[229] *Id.* at *11.

[230] Merges, *Go Ask* Alice, *supra* note 30 (emphasis added).

[231] 772 F.3d 709, 716 (Fed. Cir. 2014) (emphasis added).

[232] *See* Parker v. Flook, 437 U.S. 584, 594 (1978); *see also supra* Section I.A.

[233] *See* Ca. Inst. of Tech. v. Hughes Commc'ns Inc., No. 2:13-cv-07245-MRP-JEM, 2014 WL 5661290, at *11 (C.D. Cal. Nov. 3, 2014) (arguing that, while the point-of-novelty framework may be "a clear test of patentability," the court in *McRO* "misread the law"); Enfish, LLC v. Microsoft Corp., No. 2:12-cv-07360-MRP-MRW, 2014 WL 5661456, at *4 (C.D. Cal. Nov. 3, 2014) ("The Supreme Court did not revive *Flook*'s methodology in *Bilski*, *Mayo*, or *Alice*.").

[234] Mayo Collaborative Servs. v. Prometheus Labs., 132 S. Ct. 1289, 1298 (2012) (emphasis added) (quoting *Flook*, 437 U.S. at 590)).

term “obvious,” a term generally reserved for the question of obviousness under § 103.[235] Also in *Mayo*, the Court noted that *Diehr* was distinguishable from *Flook* because “[i]t nowhere suggested that all [the steps of the method at issue in *Diehr*], or at least the combination of those steps, were in context *obvious*, *already in use*, or purely conventional.”[236] However, in *Alice*, the Court reinterpreted *Diehr* as holding that the claims at issue there were patent-eligible “because they improved an existing technological process.”[237] Accordingly, the Court used different language in its opinion to avoid imbuing aspects of novelty or obviousness into its § 101 inquiry. Going back further, *Flook* required courts to assume that the abstract idea is within the prior art, thereby again conflating abstractness and novelty.[238] The two-step framework set forth in *Alice* requires no such step.[239] The Supreme Court in *Alice* thus has a different understanding of what constitutes “enough” and an “inventive concept” than the Court did in *Mayo* and *Flook*.

Moreover, given the Supreme Court’s careful choice of words, the phrase “well-understood, routine, conventional activities previously known to the industry” must mean something different from “novel” or “non-obvious” as required by § 102 and § 103, respectively.[240] Looking to their plain meanings, it is possible to chart the different phrases along a continuum of how much “more” a claimed element described in each manner adds to an abstract idea:

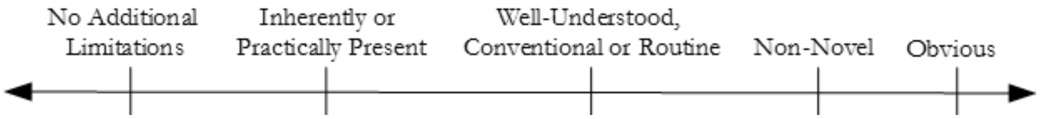
No Additional Limitations
Inherently or Practically Present
Well-Understood, Conventional or Routine
Non-Novel
Obvious

**Figure 4: Language used by courts to describe additional limitations found in a claim that covers an abstract idea, ranked from lowest to highest threshold in view of the prior art.**

Future court decisions will determine exactly what types of limitation are and, even more importantly, are not “well-understood, routine, conventional activities previously known to the industry.” However, one

[235] Alice Corp. Pty. Ltd. v. CLS Bank Int’l, 134 S. Ct. 2347 (2014).

[236] *Mayo Collaborative Servs.*, 132 S. Ct. at 1299 (emphasis added).

[237] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2358.

[238] *Flook*, 437 U.S. at 594 (1978).

[239] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2355.

[240] *See* Ca. Inst. of Tech. v. Hughes Commc’ns Inc., No. 2:13-cv-07245-MRP-JEM, 2014 WL 5661290, at *14 (C.D. Cal. Nov. 3, 2014) (“[C]onventional elements do not constitute everything in the prior art, although conventional elements and prior art may overlap”).

thing is clear—unlike in *Flook*, the Court in *Alice* did not assume that the abstract ideas found in the claims were in the prior art and then evaluate the claim for patentability. Accordingly, the "inventive concept" referenced by the Court in *Alice* does not require strict novelty or non-obviousness, and *Alice* did not revive *Flook*.

Finally, if *McRO*'s interpretation of *Alice* were correct and *Flook* has been revived, the vast majority of software patents would be invalid, with no distinction based on claim scope being possible.[241] For example, *Flook*'s framework does not allow a court to distinguish between claims 1, 9, and 10 of U.S. Patent No. 6,285,999 ("the Google page rank patent") based on claim scope.[242] This patent forms the basis of the Google search engine

[241] *Id.* at *11 (arguing that "it is difficult to imagine any software patent that survives" a revival of *Flook*'s point-of-novelty framework, since "most inventions today build on what is known in the art, and an improvement to software will almost inevitably be an algorithm or concept which, when viewed in isolation, will seem abstract."); *Flook*, 437 U.S. at 595–96 (implying that computer programs categorically do not qualify for patent protection absent congressional intervention).

[242] These claims of the Google page rank patent include the following limitations:

> 1. A computer implemented method of scoring a plurality of linked documents, comprising:
>
> obtaining a plurality of documents, at least some of the documents being linked documents, at least some of the documents being linking documents, and at least some of the documents being both linked documents and linking documents, each of the linked documents being pointed to by a link in one or more of the linking documents;
>
> assigning a score to each of the linked documents based on scores of the one or more linking documents and
>
> processing the linked documents according to their scores.

U.S. Patent No. 6,285,999 col. 8 ll. 55–67 (filed Jan. 9, 1998).

> 9. A computer implemented method of ranking a plurality of linked documents, comprising:
>
> obtaining a plurality of documents, at least some of the documents being linked documents and at least some of the documents being linking documents, at least some of the linking documents also being linked documents, each of the linked documents being pointed to by a link in one or more of the linking documents;
>
> generating an initial estimate of a rank for each of the linked documents;
>
> updating the estimate of the rank for each of the linked documents using ranks for the one or more linking documents; and
>
> processing the linked documents according to their updated ranks.

'999 Patent col. 9 l. 55–col. 10 l. 2.

and discloses an algorithm for ranking "linked documents" (e.g., web pages) called page rank.[243] In the Notice of Allowance issued during prosecution, the Examiner indicated that the prior art does not show "assigning a score to each of the linked documents based on scores of the one or more linking documents," as required by claim 1, "generating an initial estimate of a rank for each of the linked documents [and] updating the estimate of the rank for each of the linked documents using ranks for the one or more linking documents," as required by claim 9, and "automatically performing a random traversal of a plurality of linked documents [and] assigning a rank to the linked document that is dependent on the number of times the linked document has been traversed," as required by claim 10.[244] If a court were to construe each one of these limitations broadly, the court may decide that each limitation, taken independently, is a step in an algorithm and thus an abstract idea

> 10. A computer implemented method of ranking a plurality of linked documents, comprising:
>
> automatically performing a random traversal of a plurality of linked documents, the random traversal including selecting a random link to traverse in a current linked document;
>
> for each linked document that is traversed, assigning a rank to the linked document that is dependent on the number of times the linked document has been traversed; and
>
> processing the plurality of linked documents according to their rank.

'999 Patent col. 10 ll. 3–14.

[243] *See* Bill Slawski, *10 Most Important SEO Patents: Part 1—The Original PageRank Patent Application*, SEO BY THE SEA (Dec. 9, 2011), http://www.seobythesea.com/2011/12/10-most-important-seo-patents-part-1-the-original-pagerank-patent-application/; Sergey Brin & Lawrence Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, 30 COMPUTER NETWORKS AND ISDN SYS. 107 (1998), *available at* http://infolab.stanford.edu/~backrub/google.html. The page rank algorithm ranks a particular web page based on how many other web pages include links to the particular web page and the ranks of these other web pages—thereby ranking pages based on the probability that a web surfer ends up on any given web page. *See* '999 Patent col. 8 ll. 57–67, col. 3 ll. 4–16 (filed Jan. 9, 1998). This invention located more relevant and thus better search results for a user than the leading commercial search engines of the time. Brin, *supra*. Even today, when Google Search incorporates a large number of additional factors to help determine a web page's rank, the page rank algorithm still plays a vital role. Ray Comstock, *So... You Think SEO Has Changed?*, SEARCH ENGINE WATCH (Mar. 19, 2014), http://searchenginewatch.com/sew/opinion/2334934/so-you-think-seo-has-changed. At least one commentator has used the Google page rank patent as an example of a software patent that "ought to survive Section 101 analysis." Merges, *Go Ask* Alice, *supra* note 30.

[244] '999 Patent Notice of Allowance dated April 22, 2001.

under *Benson*.[245] Thus, accepting that these three limitations actually constitute the point of novelty of claims 1, 9 and 10, respectively, a court employing *Flook*'s point-of-novelty framework would hold that claims 1, 9 and 10 are patent-ineligible. Accordingly, even though these claims of the Google page rank patent have different breadths and preemptive effects, and thus ought to be treated differently under the substantive limitation theory of the purpose of the patent-eligible subject matter exceptions, *Flook*'s framework does not allow a court to make such a distinction between narrowly tailored and overly broad claims on software innovation.

### 2. Only Physical Limitations can be "Enough"

Another possible interpretation of *Alice*, explicitly relied on in the Federal Circuit's *buySAFE, Inc. v. Google, Inc.* decision, is that an "inventive concept" must be "in the *physical* realm of things and acts—a 'new and useful application' of the ineligible matter in the *physical* realm—that ensures the patent covers something 'significantly more than' the ineligible subject matter."[246] Similarly, in *Digitech Image Technologies, LLC v. Electronics for Imaging, Inc.*, the Federal Circuit found that the patent at issue "claims an abstract idea because it describes a process of organizing information through mathematical correlations and is not tied to a specific *structure or machine*."[247]

Additionally, in view of *Alice*, any physical limitation must be more than a "wholly generic computer implementation" to confer patent eligibility.[248] This requirement has led some courts to conclude that a computer must do more than calculations of a type previously performed by humans before it is "enough."[249] This conclusion echoes Judge Lourie's *CLS Bank* opinion regarding computer implementation—"At its most basic, a computer is just a calculator capable of performing mental steps faster than a human could. Unless the claims require a computer to

[245] *See* '999 Patent col. 8 ll. 54–67, col. 9 l. 55–col. 10 l. 14. This broad construction is used for illustrative purposes only. A court may also decide that a narrower construction is warranted, and that these steps are not patent-ineligible abstract ideas.

[246] 765 F.3d 1350, 1353 (Fed. Cir. 2014) (emphasis added) (quoting *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2355 (2014)).

[247] 758 F.3d 1344, 1350 (Fed. Cir. 2014) (emphasis added).

[248] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2358.

[249] *See* Helios Software, LLC v. SpectorSoft Corp., No. 12-081-LPS, 2014 WL 4796111, at *17 (D. Del. Sept. 18, 2014) (concluding that, while a wholly generic computer implementation cannot confer patent eligibility, a claimed computer performing steps that "could [not] be performed by a human alone" is a "meaningful limitation" sufficient for patent eligibility).

perform operations that are not merely accelerated calculations, a computer does not itself confer patent eligibility."[250]

Courts and commentators frequently combine the physical limitation requirement with the pen-and-paper test.[251] This causes decisions and articles to be inconsistent when discussing whether steps that humans cannot perform are themselves "enough" to confer patent eligibility, or if the recitation of a physical computer is "enough" because such steps constitute a non-generic use of the computer or meaningfully tie the steps to the computer. For example, the court in *Helios Software LLC* found claims patent-eligible because they "sufficiently tie the claimed method to a machine" by including "limitations [that] could [not] be performed by a human alone."[252] As another example, Professor Merges has argued that using hardware elements in a novel manner is potentially "enough" to confer patent eligibility.[253] However, as a counterexample, the court in *Card Verification Solutions, LLC v. Citigroup, Inc.* indicated that claims that use a computer only for "manipulating, reorganizing, or collecting data" are not patent-eligible, but held that claims reciting "pseudorandom tag generating software" that cannot be mimicked "by a human with nothing more than pen and paper" are.[254] Thus, in that case, the software itself, not the underlying hardware, was "enough" to confer patent eligibility.[255]

*Alice* does not state or imply that only physical limitations are "enough" to confer patent eligibility. Instead, by listing in dicta what the additional limitation in the claims at issue *did not* do and thus *did not* confer patent eligibility, the Court implied that if a limitation *did* do those things it *would* confer patent eligibility.[256] Thus, the Court first implied that steps that are *not* "well-understood, routine, conventional activit[ies] previously known to the industry" *are* "enough."[257] Second, the Court implied that a claim that "improve[s] the functioning of the computer itself" or "effect[s] an improvement in any other technology or technical field" is patent-eligible.[258] Unfortunately, it is unclear what the Court meant by

[250] *CLS Bank Int'l*, 717 F.3d at 1286 (Lourie, J. concurring).

[251] *See Helios Software, LLC*, 2014 WL 4796111, at *17; *supra* Section IV.A.2.

[252] *Helios Software, LLC*, 2014 WL 4796111, at *17.

[253] Merges, *Go Ask* Alice, *supra* note 30.

[254] No. 13 C 6339, 2014 WL 4922524, at *4–5 (N.D. Ill. Sept. 29, 2014).

[255] *See id.*

[256] *See* Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2359–60 (2014).

[257] *See id.* at 2359 (internal quotation marks omitted) (quoting *Mayo Collaborative Servs. v. Prometheus Labs.*, 132 S. Ct. 1289, 1294 (2012)).

[258] *See id.* at 2359–60. This was the government's position on limitations that are "enough" for patent eligibility. The government's amicus curiae brief lists "an improvement in the functioning of the computer as a computer, e.g., by making it more

"improv[ing] the functioning of the computer itself,"[259] what types of limitations the Court considered "[an]other technology or technical

efficient[,]" as one of its factors that support a finding of patent eligibility. Brief for the United States as Amicus Curiae in Support of Respondents at 30, *Alice Corp. Pty. Ltd.*, 134 S. Ct. 2347 (No. 13-298). While the government's brief does not discuss whether claims similarly drawn to abstract ideas that "effect an improvement in any other technology or technical field" are patent-eligible, the government took this position during oral arguments. Transcript of Oral Argument, *supra* note 121, at 45 ("Any software patent that improves—that is used to improve another technology is eligible."). It remains to be seen to what extent the Supreme Court adopted the government's underlying logic and accompanying arguments in addition to the government's final conclusion.

[259] Allowing patents on software that "improve[s] the functioning of the computer itself" may lead to overly broad patents that have a much larger preemptive effect than the patents at issue in *Alice* or any other business method patent. For example, the Fast Fourier Transform, when implemented on a computer, converts a signal from the time domain to the frequency domain, and vice versa, much faster than an earlier algorithm used for this purpose, and thus "improve[s] the functioning of the computer itself." *See supra* note 186; Eric W. Weisstein, *Fast Fourier Transform*, MATHWORLD—A WOLFRAM WEB RESOURCE, http://mathworld.wolfram.com/FastFourierTransform.html (last visited Feb. 8, 2015). Based on the Supreme Court's dicta, a claim directed to the Fast Fourier Transform implemented on a computer might therefore be patent-eligible. *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2359. However, such a claim would grant a monopoly on an important tool in a practitioner's repertoire. *See, e.g.*, Newell, *supra* note 17, at 1028. While it is debatable whether the Fast Fourier Transform truly qualifies as a basic tool[] of scientific and technological work," since practitioners would still be free to use the older and slower algorithm to accomplish the same result, at the very least a patent on the Fast Fourier Transform would present an equivalent preemption issue to the patent found patent-ineligible in *Benson*. Accordingly, given the right software example, the Supreme Court's dicta that an "improve[ment to] the functioning of the computer itself" confers patent eligibility may be in conflict with earlier Supreme Court precedent, especially *Benson*. Moreover, software that enables a computer to perform a new function, even if this function is a business method, literally "improve[s] the functioning of the computer itself"—the computer is able to provide a service it previously was unable to. *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2359; *In re* Alappat, 33 F.3d 1526, 1545 (Fed. Cir. 1994) ("[The Federal Circuit] held that [a new software program] creates a new machine, because a general-purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software."). At the same time, this reasoning cannot be taken too far, since otherwise even the claims at issue in *Alice* would be patent-eligible for enabling a computer to provide the new function of "intermediated settlement, i.e., the use of a third-party to mitigate settlement risk." *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2356. In the end, clarifying this point may only affect software patents whose *only* purpose is facilitating a business method, since even the government's position was that "a claim that discloses software that enables a computer to manipulate data in an innovative way would be patent-eligible even if its *primary* utility is in making financial transactions more efficient." Brief for the United States as Amicus Curiae in Support of Respondents at 32, Alice Corp. Pty. Ltd., 134 S. Ct. 2347 (No. 13-298) (emphasis added).

field,"[260] or what the Court meant by "effect[ing] an improvement" in this other technology or technical field.[261] However, one thing is clear—*Alice* does not indicate that the "well-understood, routine, conventional activities" or the "other technology or technical field" must be in the physical realm. In fact, the Supreme Court previously used the phrase "emerging *technologies*" to describe Information Age inventions that are potentially not "grounded in a physical or other tangible form," thereby indicating that an improvement in another technology does not require an improvement in, or any other tie to, the physical realm.[262]

Additionally, a physical limitation is not necessary to achieve the purpose of the patent eligibility exceptions. Specifically, the focus on physical limitations echoes Chief Judge Rader's coarse filter perspective from *CLS Bank* that a claim must "meaningfully tie [an] idea to a *concrete reality* or actual application" in order to be patent-eligible.[263] However,

[260] Specifically, it is unclear whether enabling or facilitating a business method constitutes "an improvement in any other technology or technical field." On one hand, the court's use of the term "technology or technical field" harks back to the distinction drawn between business innovation and technological innovation by the concurring opinion in *Bilski*. *See* Bilski v. Kappos, 130 S. Ct. 3218, 3254 (2010) (Stevens, J., concurring) ("Business innovation, moreover, generally does not entail the same kinds of risk as does more traditional, technological innovation."). Additionally, during oral arguments, the government took the position that an improvement to "a human activity," such as "finance or something in business," does not constitute "an improvement in any other technology or technical field." Transcript of Oral Argument, *supra* note 121, at 48. At the same time, the majority in *Bilski* explicitly refused to make such a distinction between business and technology innovation, arguing that drawing a line between the two would be difficult, since "technologies" include "technologies for conducting business more efficiently." *See Bilski*, 130 S. Ct. at 3228. Drawing the same line in the context of identifying whether an "inventive concept" is present would be contrary to the spirit of this holding. *But see* I/P Engine, Inc. v. AOL Inc., 576 F. App'x 982, 982 (Fed. Cir. 2014) (Mayer, J. dissenting) ("The Supreme Court in *Alice* . . . , for all intents and purposes, recited a 'technological arts' test for patent eligibility.").

[261] The Supreme Court's dicta does not specify what type and level of connection must exist between the claimed patent-ineligible subject matter and another "technology or technical field" for the claimed subject matter to "effect an improvement" in this other "technology or technical field." *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2359. After all, merely reciting this other "technology or technical field" in the claim would, without more, presumably be no more than "limiting the use of [the patent-ineligible subject matter] 'to a particular technological environment,'" and thus insufficient to confer patent eligibility. *See id.* at 2358–59.

[262] *See Bilski*, 130 S. Ct. at 3226–27 (emphasis added).

[263] *See* CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1299–1300 (Fed. Cir. 2013) (en banc) (Rader, C.J., concurring in part and dissenting in part) (emphasis added), *aff'd*, 134 S. Ct. 2347 (2014); *see also id.* at 1320 (Moore, J., dissenting in part) (arguing that the system claims at issue in *CLS Bank* cannot be directed to an abstract

*Alice* does not endorse that view.[264] Instead, *Alice* states that "[t]he fact that a computer 'necessarily exist[s] in the physical, rather than purely conceptual, realm,' is beside the point."[265] While this statement addresses the petitioner's argument that a physical computer cannot be an abstract idea, it also cuts the other way—whether a claim is tied to the "physical . . . realm" is not the focus of the § 101 inquiry; it is "beside[s] the point."[266] After all, even an additional limitation that is not grounded in the "physical . . . realm" can "ensure that the patent in practice amounts to significantly more than a patent upon [an abstract idea]."[267] Additionally, the relevant details in software that explain how a particular user experience is achieved and that thus limit a claim on the software to what a practitioner actually built are usually not found in any non-generic piece of hardware, but in a particular implementation or a particular set of libraries.[268] Both of these are additional algorithms and not in the "physical realm," yet each is a more meaningful limitation on claim scope than a recitation of hardware.[269] Finally, implementing a particular feature in hardware (i.e., the physical realm) or software (i.e., an abstract idea) is merely a design choice and thus not indicative of the preemptive effect of a patent claiming this feature.[270]

Moreover, requiring that software claims include physical limitations beyond a general-purpose computer is inconsistent with an important goal in software development—independence from any specific hardware platform.[271] To achieve this goal, most software is designed to be hardware agnostic and thus rarely uses the underlying hardware beyond the functions available on most general-purpose computers—i.e., calculations a human, alone or with pen and paper, can perform.[272] Thus, few advances in software rely on the underlying hardware for non-generic function that

idea, since these claims cover "a tangible item that you could pick up and put on your desk.").

[264] *See supra* Section II.B.

[265] *Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2358 (citations omitted).

[266] *See id.*

[267] *See id.* at 2355 (internal quotation marks omitted) (quoting *Mayo Collaborative Servs. v. Prometheus Labs.*, 132 S. Ct. 1289, 1294 (2012)).

[268] *See supra* Section III.A.

[269] *See id.*

[270] *See id.*

[271] *See id.*

[272] *See id.*; CLS Bank Int'l v. Alice Corp. Pty. Ltd., 717 F.3d 1269, 1306 n.7 (Fed. Cir. 2013) (en banc) (Rader, C.J., concurring in part and dissenting in part), *aff'd*, 134 S. Ct. 2347 (2014).

cannot be described in terms equivalent to human mental steps.[273] At the very least, well-designed software that performs a particular function using a general-purpose computer should be equally patent eligible as software that performs the same function, but relies on specialized hardware and is thus inferior.

Finally, requiring a physical limitation does not allow courts to distinguish software patents based on claim scope. Narrowly tailored claims on innovative software running on a general-purpose computer, and overly broad claims on business methods running on the same general-purpose computer, are equally patent-ineligible. For example, if a court construed claims 1, 9, and 10 of the Google page rank patent[274] broadly, the court might decide that the only recitation of hardware or other physical element is the recitation of "computer implementation."[275] Furthermore, the steps in each of the claims can be performed by a human (e.g., a human could rank a set of research papers that cite to each other in the claimed manner) and are thus not enough to ensure that the recitation of a "computer" amounts to more than "generic computer implementation."[276] At the same time, claim 1 is much broader in scope than claims 9 and 10. Claim 1 in essence only requires that web pages be ranked according to a particular mathematical formula disclosed in the Google page rank patent,[277] whereas claims 9 and 10 specify how this ranking is to be accomplished.[278] However, if only physical limitations are "enough" to confer patent eligibility, the court would be unable to differentiate between broader claim 1 and narrower claims 9 and 10, and might instead hold that all claims of the Google page rank patent are equally invalid, even though the different claims have different scopes and thus different preemptive effects. Accordingly, requiring a physical limitation suffers from the same defect as the related pen-and-paper test—it limits patent eligibility categorically, instead of allowing courts to

[273] *See* CLS Bank Int'l, 717 F.3d at 1306 n.7 (Rader, C.J., concurring in part and dissenting in part).

[274] *See supra* note 242.

[275] *See* U.S. Patent No. 6,285,999 col. 8 ll. 55–67, col. 9 l. 54–col. 10 l. 14 (filed Jan. 9, 1998). This broad construction is for illustrative purposes only. A court may also construe the claim in a manner that include additional ties to the physical realm.

[276] *See* '999 Patent col. 8 ll. 55–67, col. 9 l. 54–col. 10 l. 14. Once again, this Note assumes an overly broad construction of the claims of the Google page rank patent for illustrative purposes.

[277] *See* '999 Patent col. 4 ll. 15–25, col. 8 ll. 55–67, col. 9 l. 54–col. 10 l. 14.

[278] *See* '999 Patent col. 9 l. 54–col. 10 l. 14.

conduct a nuanced analysis based on claim scope to identify and invalidate overly broad claims.[279]

### 3. A Second Abstract Idea is "Enough"

Instead of employing *Flook*'s point-of-novelty framework or demanding that a claim include additional physical limitations, this Note proposes that courts identify multiple separate abstract ideas within a single claim (i.e., apply the proposed multiple abstract ideas test from Section IV.A.3), and then inquire for each whether the other abstract ideas are "enough" to confer patent eligibility. Since *Alice*, at least one lower court has already acknowledged that "mathematical operation[s]" can be "enough" for patent eligibility, but did not address why these "mathematical operation[s]," themselves, ought to be patent-eligible.[280] The proposed approach integrates the argument that the details of an algorithm can confer patent eligibility into the *Alice* framework.[281]

For purposes of discussion, consider a claim that recites a particular user experience (i.e., *what* the claimed software accomplishes) and a particular implementation that provides the user experience (i.e., *how* the claimed software accomplishes this):

[279] *See supra* Section IV.A.2.

[280] *See* Ca. Inst. of Tech. v. Hughes Commc'ns Inc., No. 2:13-cv-07245-MRP-JEM, 2014 WL 5661290, at *18 (C.D. Ca. Nov. 3, 2014) (holding that "mathematical operations," such as "a linear transform operation to produce L transformed bits and the accumulation of these bits to produce a codeword," can be inventive concepts sufficient to confer patent eligibility). Other courts noted that additional software limitations can also confer patent eligibility, but have shied away from describing these additional limitations using terminology associated with abstract ideas. *See* DDR Holdings, LLC v. Hotels.com L.P., 773 F.3d 1245, 1259 (Fed. Cir. 2014) (claims that recite "a specific way to automate the creation of a composite web page . . . that incorporates elements from multiple sources" are patent-eligible); Enfish, LLC v. Microsoft Corp., No. 2:12-cv-07360-MRP-MRW, 2014 WL 5661456, at *12 (C.D. Cal. Nov. 3, 2014) (discussing that a hypothetical chess program that uses dynamic memory allocation while playing a game of chess would be patent-eligible); Card Verification Solutions, LLC v. Citigroup, Inc., No. 13 C 6339, 2014 WL 4922524, at *4 (N.D. Ill. Sept. 29, 2014) (finding a claim reciting "pseudorandom tag generating software" to be patent-eligible).

[281] *See Ca. Inst. of Tech.*, 2014 WL 5661290, at *15–19; Collins, *supra* note 174, at 1402. Additionally, claims found valid under the proposed approach would explicitly include limitations that others have proposed ought to be imported into software claims by interpreting them as means-plus-function claims. *See* Lemley, *The Return of Functional Claiming*, *supra* note 175, at 909.

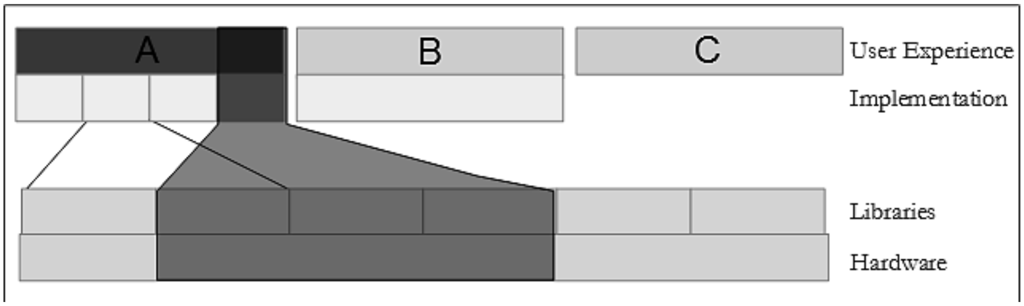

**Figure 5: The scope of a claim reciting a user experience and an implementation is represented by the transparent overlay.**

In the context of software, it is likely that both the user experience and the implementation are abstract ideas.[282] A court should first evaluate the claim from the perspective of the user experience, and determine whether the additional limitations (here, the recitation of a particular implementation) are "enough" to confer patent eligibility. If the point-of-novelty of the invention lies in the implementation, then by definition the recited implementation does not constitute "well-understood, routine, conventional activities" and therefore is "enough."[283] If the point-of-novelty lies in the user experience, so long as the claim recites an actual implementation and not just generic steps inherent to the user experience, the implementation that accomplishes this novel user experience presumably still constitutes more than "well-understood, routine, conventional activities" and is therefore "enough."[284]

Now, there might be some novel user experiences, such as user experience *B*, which at the time a patent is litigated can only be implemented in a single manner. However, since additional alternative implementations may be discovered in the future, unless a defendant can show that such alternatives are impossible, a court should still find a claim directed to user experience *B* and the underlying implementation patent-eligible:

> [W]e must be wary of facile arguments that a patent preempts all applications of an idea. It may often be easier for an infringer to argue that a patent fails § 101 than to figure out a different way to implement an idea, especially a way that is less complicated—less liable to get out of order—less expensive in construction, and its operation. But the patent law does not privilege the leisure of

[282] *See supra* Section III.A; *supra* Section IV.A.1.

[283] *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2359; *supra* Section IV.B.1.

[284] *See id.*; *see also supra* Section III.A.

> an infringer over the labors of an inventor. Patents should not be casually discarded as failing § 101 just because the infringer would prefer to avoid the work required to develop non-infringing uses of the abstract idea at the heart of an appropriately circumscribed invention.[285]

There may also be novel user experiences, such as user experience *C*, whose implementations are either inherent in the recitation of the user experience itself, or so straightforward as to be no more than "well-understood, routine, conventional activities."[286] In this scenario, no matter how willing a practitioner is to narrow her claim, patent protection is simply not available. The claims in *Alice* fall into this category—receiving information regarding transactions, keeping track of these transactions, and communicating instructions are all such straightforward steps that it is difficult to imagine any additional implementation details that are not inherent in their recitation.[287] However, if no complex implementation is needed to achieve a novel user experience, patent protection may very well be undesirable from a social perspective. First, it is probably impractical to try and keep a user experience (e.g., the user interface of a word processing program) a secret.[288] Accordingly, trade secret protection is not available, and patent protection is not needed to compel disclosure of the novel features of the invention.[289] Second, if the user experience is truly simple to implement, which is frequently the case with business methods, then no large investments in research or development are required to bring it to market, which in turn means that patent protection (and the monopoly resulting therefrom) is not necessary to allow the practitioner to recoup her investment.[290] Thus, some potentially novel inventions, such as the

[285] McRO, Inc. v. Activision Publ'g, Inc., No. CV 14-336-GW(FFMx), 2014 WL 4759953, at *7 (C.D. Cal. Sept. 22, 2014) (citations and internal quotation marks omitted).

[286] *See* Ultramercial, Inc. v. Hulu LLC, 722 F.3d 1335, 714–15 (Fed. Cir. 2014) (even accepting that the abstract idea at issue may be novel, the court concluded that the claim did not include an inventive concept sufficient to confer patent eligibility).

[287] *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2359.

[288] *See* Mobius Med. Sys., LP v Sun Nuclear Corp., No. 4:13-CV-3182, 2013 WL 6498981, at *12–13 (S.D. Tex. Dec. 10, 2013) (finding that user interfaces (i.e., user experiences) cannot be protected as trade secrets, but that algorithms used to generate these user interfaces (i.e., implementations) can be protected as trade secrets, since the parties' distribution agreement forbade decompiling, disassembling, or reverse engineering of these algorithms).

[289] *See id.*; Graham, *supra* note 168 (absent patent protection, practitioners will resort to trade secrecy to protect their innovation).

[290] *See* Bilski v. Kappos, 561 U.S. 593, 652 (2010) (Stevens, J., concurring) (Business innovation "generally does not require the same enormous costs in time,

alleged innovation reflected in the *Alice* patents and other pure business method patents, remain patent-ineligible under the proposed approach.

After determining that a claim contains "enough" additional limitations to confer patent eligibility onto an abstract user experience, a court should repeat the process from the perspective of the equally abstract implementation. Now, the additional limitation becomes the user experience. By definition, an implementation enables or "effects an improvement" upon the user experience, which Supreme Court dicta seems to suggest is "enough" for patent eligibility as long as the user experience falls in "any other technology or technological area."[291] Accordingly, from the perspective of the implementation layer, the hypothetical claim includes an inventive concept sufficient for patent eligibility.

This proposed approach is consistent with the purpose of the patent-eligible subject matter exceptions endorsed by the Supreme Court—it ensures that claims contain substantive limitations and are not overly broad.[292] Specifically, a software claim that recites both *what* it accomplishes and *how* it accomplishes this result ensures that the scope of patent protection is commensurate with what a practitioner actually developed.[293] Thus, in the example above, the practitioner would be granted a claim that only covers the particular implementation she designed, and others would be free to attempt to provide the same user experience using different implementations.[294]

As another example, if the practitioner designed a novel and useful library, the resultant claim would only be valid under the proposed approach if it also recites a user experience or implementation that the practitioner identified as benefiting from and thus being improved by this library. For purposes of discussion, consider a claim that recites a particular library and a particular user experience:

research, and development, and thus does not require the same kind of compensation to innovators for their labor, toil, and expense." (citations and internal quotations omitted)).

[291] *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2359; *supra* note 260; *supra* note 261.

[292] *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2354; Lemley, *Life After* Bilski, *supra* note 101, at 1316–17; *supra* Section II.B.

[293] *See* Lemley, *Life After* Bilski, *supra* note 101, at 1316–17; *supra* Section III.B.

[294] *Compare supra* Figure 1 *with supra* Figure 5.

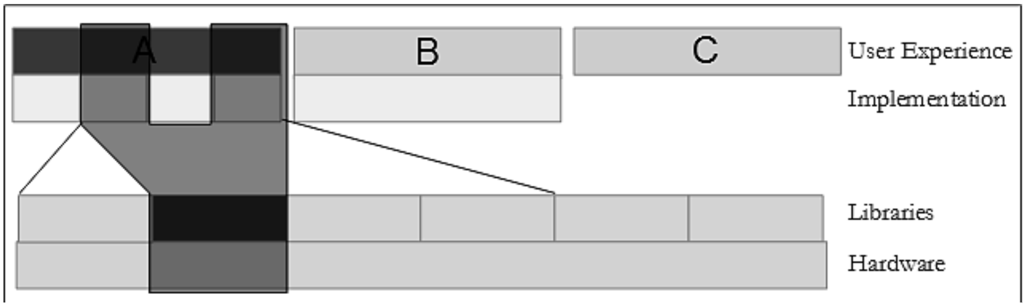

**Figure 6: The scope of a claim reciting a user experience and a library is represented by the transparent overlay.**

Such a claim would be valid under the proposed approach because the library would constitute more than "well-understood, routine, conventional activities" and thus confer patent eligibility onto the user experience, while the user experience would be a technology or technical area improved upon by the library and thus confer patent eligibility onto the library. [295] The resultant claim would also not be overly broad, since its scope is limited to user experiences and implementations that actually benefited from the inventor's contribution and that the inventor actually identified, as opposed to extending into fields the inventor did not even consider.[296]

Additionally, other practitioners have more notice of a claim directed to a library, if the claim also recites a user experience or implementation that benefits from this library. Under the layered approach to software design, another practitioner working on a particular implementation does not care and does not have any reason to know how a particular library functions, unless she happens to have designed the library herself.[297] This other practitioner can only perform a freedom to operate search based on the information actually available to her—what she is currently working on (i.e., the implementation) and what she hopes to accomplish (i.e., the user experience). Additionally, if the other practitioner did design the library herself, the proposed approach to patent eligibility would allow her to confine her freedom to operate search to the field she is actually using this library for (i.e., the implementation and user experience her library is meant to facilitate). Either way, the proposed approach ensures that a patent directed to a new and useful library is only valid if other

[295] *See Alice Corp. Pty. Ltd.*, 134 S. Ct. at 2359; *supra* note 260; *supra* note 261.
[296] *See* Lemley, *Life After* Bilski, *supra* note 101, at 1316–17; *supra* Section III.B.
[297] *See supra* Section III.A.

practitioners have a mechanism for finding it during a reasonably limited freedom to operate search.[298]

Finally, the proposed approach to § 101, unlike *Flook*'s framework or the requirement of a physical limitation, allows courts to distinguish software patents that are commensurate in claim scope with the inventor's contribution to the field from overly broad patents. For example, claim 1 of Google's page rank patent[299] is probably invalid even under the proposed approach, since it merely recites a user experience (i.e., ranking web pages according to a particular mathematical formula) without actually specifying any implementation or library that allows this user experience to be achieved.[300] In contrast, claims 9 and 10,[301] in addition to the abstract idea of ranking web pages, each recite one of two alternative implementations, neither of which is merely "well-understood, routine, conventional activities" and that thus each confer patent eligibility onto the abstract mathematical ranking formula.[302] At the same time, each of these two alternative implementations enables and thus "effect[s] an improvement" in the ranking of web-pages according to the mathematical formula, so the ranking of web pages in turn confers patent eligibility onto these two implementation limitations.[303] Accordingly, under the proposed approach, Google would still have valid claims 9 and 10 covering two particular implementations that rank web pages according to the mathematical formula it invented, but would be unable to monopolize the mathematical formula itself via overly broad claim 1. Thus, the proposed approach ensures that not all claims of every software patent are blindly lumped together with overly broad pure business method patents and summarily rendered invalid; instead, it allows courts to distinguish between software claims based on scope.

The proposed approach to identifying an "inventive concept" that is "enough" to confer patent eligibility onto abstract ideas stays true to

[298] *See supra* Section III.B.

[299] *See supra* note 242.

[300] *See* U.S. Patent No. 6,285,999 col. 4 ll. 15–25, col. 8 ll. 54–67 (filed Jan. 9, 1998). This conclusion assumes for illustrative purposes that the abstract idea encompassed in claim 1 is the specific mathematical formula disclosed in the Google page rank patent. A potentially more appropriate definition of the abstract idea is the broader concept of ranking web pages. *See id.* col. 1 ll. 19–22. Additionally, this Note assumes an overly broad construction of claim 1 for illustrative purposes. If either the abstract idea is defined more broadly or the claim is construed more narrowly, claim 1 of the Google page rank patent is likely valid under the proposed approach.

[301] *See supra* note 242.

[302] *See* '999 Patent col. 9 l. 55–col. 10 l. 14.

[303] *See id.*; *supra* note 261.

Supreme Court precedent, including the *Alice* and *Benson* decisions, and accomplishes the goal behind § 101 jurisprudence of serving as a substantive limitation that protects against overly broad patents. Most importantly, and unlike the other two approaches to determining whether a claim adds "enough" to an abstract idea, the proposed approach allows courts to distinguish between patents on software running on a general-purpose computer based on claim scope.

## V. CONCLUSION

*Alice* cemented § 101 as a substantive limitation on patent scope, one whose purpose is to protect against overly broad patents. Under *Alice*, one cannot obtain a patent by describing a common business method and instructing a practitioner to implement it with a computer. At the same time, at least some software remains patent-eligible. Thus, courts need a mechanism to distinguish between more desirable software patents and those that are less desirable. Given the purpose of the patent-eligible subject matter exceptions, any such approach needs to distinguish between software patents that are consummate in scope with an inventor's contribution and those that are overly broad, instead of categorically rendering all software patents invalid.

The proposed solution accepts that a single piece of software includes many layers, each of which constitutes a separate patent-ineligible algorithm. These different algorithms, although abstract ideas when viewed individually, should be "enough" to confer patent eligibility to each other if they either improve or enable each other. Thus, a claim that recites a user experience (i.e., what a piece of software accomplishes) and an implementation or a library that is more than "well-understood, routine, conventional activit[y]" and that enables or improves the user experience should be patent-eligible. This allows courts to distinguish between a narrowly tailored software patent and an overly broad business method patent when evaluating patent eligibility.